

Université Paris Sud-11

Mémoire

présenté pour obtenir

le diplôme d'Habilitation à Diriger les Recherches

présenté par

David ROUSSEAU

Sujet :

Recherche du «Petit Higgs» et
développement des algorithmes de reconstruction dans ATLAS

Soutenu le 4 Mai 2007 devant le jury composé de :

Alain Bonissent Rapporteur

Aldo Deandrea Rapporteur

Daniel Fournier

Albert de Roeck Rapporteur

Guy Wormser Président

AVERTISSEMENT

Les travaux décrits dans ce mémoire d'habilitation ont été réalisés au cours des années 2000 à 2006 environ. Par souci de présenter un document cohérent et actuel, mes activités de 1992 (date de ma thèse) à 2000, n'ont pas été décrites. Il s'agissait principalement de :

- avec ALEPH au LEP, mesure de V_{cb} et des facteurs de forme dans les désintégrations semi-leptoniques exclusive $B^0 \rightarrow D^{*+} \ell^- \nu$ et $B^0 \rightarrow D^+ \ell^- \nu$
- avec ALEPH au LEP, recherche du méson B_c
- alignement du détecteur de vertex en Silicium d'ALEPH par une méthode dite «de la grande matrice»
- étude de la violation de CP dans les désintégrations hadroniques à deux corps des mésons beaux dans Atlas
- nombreuses études de performance pour les TDR d'Atlas

TABLE DES MATIÈRES

Avertissement	1
Table des matières	2
Introduction	4
1 Développements informatiques d’Atlas	5
1.1 Environnement et cycle de développement	5
1.1.1 Développements chronologiques	5
1.1.2 Intégration et tests de validation	7
1.1.3 l’eXtreme Programming	10
1.2 Architecture	12
1.2.1 Orienté objet, mais pas trop	12
1.2.2 Séparation données-algorithmes	13
1.2.3 Event Data Model de la simulation et des données brutes	15
1.3 Structure des algorithmes de reconstruction	16
1.3.1 Survol	17
1.3.2 Des Algorithmes aux Tools	17
1.3.3 Calorimétrie et trajectographie	21
1.3.4 Classe cinématique de base	21
1.4 Données persistantes	29
1.4.1 Ntuple combiné (CBNT)	29
1.4.2 Vers des objets persistants	31
1.4.3 Eléments du computing model	32
1.4.4 Requis généraux	33
1.4.5 Contraintes du framework	35
1.4.6 Définition des AOD et ESD	35
1.5 La grille	37
2 Algorithmes de reconstruction pour Atlas	39
2.1 Reconstruire ou la résolution d’un problème inverse	39
2.2 Atlas dans une coquille de noix	40
2.3 Empilement	40
2.4 Tracking interne	43
2.4.1 Développements récents et en cours	43
2.5 Vertexing	44
2.5.1 Ajustement du vertex	44
2.5.2 Assignment des traces aux vertex	45
2.5.3 Développements récents et futurs	46

2.6	Identification des électrons et photons	46
2.6.1	Développements en cours et futurs	47
2.7	Identification et reconstruction des muons	48
2.7.1	Développements en cours et futurs	50
2.8	Reconstruction des jets	50
2.8.1	Développements en cours et futurs	50
2.9	Energie transverse manquante	51
2.9.1	Développements en cours et futurs	51
2.10	Etiquetage des b-jet	51
2.10.1	Développements en cours et futurs	52
2.11	Identification des τ	52
3	Recherche du Petit Higgs dans Atlas	53
3.1	Recherche du quark lourd T	53
3.1.1	Etude de la désintégration $T \rightarrow Zt$	54
3.1.2	Etude de la désintégration $T \rightarrow Wb$	56
3.1.3	Etude de la désintégration $T \rightarrow Ht$	56
3.2	Recherche de nouveaux bosons de jauge	58
3.2.1	Section efficace et rapports d'embranchement	58
3.2.2	Le boson de jauge A_H	65
3.2.3	Découverte de Z_H , A_H et W_H	66
3.2.4	Observation de $Z_H \rightarrow ZH$, $A_H \rightarrow ZH$ et $W_H \rightarrow WH$ pour $m_H = 120$ GeV . .	67
3.2.5	Observation de $Z_H \rightarrow ZH$ et $W_H \rightarrow WH$ pour $m_H = 200$ GeV	77
3.2.6	Recherche des désintégrations hadroniques des Z_H and W_H	88
3.2.7	Généralisation	89
3.3	Recherche de ϕ^{++}	93
3.4	Contraintes sur le modèle, conclusion et perspectives	94
	Conclusion	97
	Glossaire	98
	Bibliographie	99

INTRODUCTION

Le LHC produira ses premières collisions avec un premier run à basse énergie dès la fin de l'année 2007. Cette année marque donc la fin du commencement, plus de 23 ans après les toutes premières études sur le LHC lancées en 1984, et 15 ans après le dépôt de la Lettre d'Intention d'Atlas marquant sa naissance. Dès le début des codes de reconstruction ont été écrits, dans le but de valider la conception du détecteur, d'évaluer ses performances et de permettre certaines études de physique en simulation complète. Cette première phase s'est achevée avec l'écriture du Physics TDR, pour lequel un code de reconstruction complet en langage fortran Atrecon était disponible. Une deuxième phase a démarré par la migration des algorithmes fortran dans un framework C++, dans le but cette fois de fournir le programme de reconstruction qui servira aux prises de données. La première partie de ce mémoire décrit les différents développements généraux qui ont été nécessaire pour arriver aux framework et utilitaires actuels. La deuxième partie passe en revue les différents algorithmes de reconstruction existant, sans tenter de présenter les performances d'une façon exhaustive, mais en indiquant les points clés et les pistes de développements actuels et futurs.

Le détecteur Atlas a été optimisé autour de quelque canaux de physique clés qui ont été étudiés avec soin pendant toute la période de développement : en particulier $H \rightarrow \gamma\gamma$ pour les photons, $H \rightarrow ZZ \rightarrow e^+e^-e^+e^-$ pour les électrons, $H \rightarrow ZZ \rightarrow \mu^+\mu^-\mu^+\mu^-$, $A \rightarrow \tau^+\tau^-$ pour l'énergie transverse manquante et l'identification des τ , $ttH \rightarrow b\bar{b}$ pour l'étiquetage des b-jets et la reconstruction des jets. Des nouveaux canaux de physique qui n'avaient pas été envisagés au départ (car issus de nouveaux développements théoriques) ont pu être étudiés récemment et démontrer la polyvalence du détecteur Atlas. Je m'attacherai dans la troisième partie au modèle dit du «Petit Higgs», dont les signatures potentielles dans Atlas sont variées.

1 DÉVELOPPEMENTS INFORMATIQUES D'ATLAS

Le premier logiciel de reconstruction d'Atlas a été réalisé en fortran pour la conception et l'optimisation du détecteur à partir du début des années 1990, culminant avec la rédaction du Physics TDR [1]. A partir de 1999, le code a été porté vers C++ et largement amélioré. Après un rappel chronologique, ce chapitre traite des aspects les plus importants du framework (au sens large, c.à.d toute l'infrastructure et les outils communs) dans lequel s'insère les développements des algorithmes de reconstruction qui seront décrit dans le chapitre suivant.

1.1 Environnement et cycle de développement

1.1.1 Développements chronologiques

Après la rédaction du Physics TDR en 1999 où toutes les études en simulation complètes ont été réalisées avec le logiciel fortran *atrecon*, l'effort s'est porté sur la transition vers C++ qui permettra de traiter la complexité (en particulier : alignements, étalonnage et filtrage en ligne) et le volume des futures données réelles et simulées. La transition vers la reconstruction C++ s'est faite adiabatiquement (i) pour préserver l'expertise accumulée dans *atrecon*, (ii) pour garder une reconstruction fonctionnelle à tout instant pour pouvoir valider les dernières évolutions de la conception du détecteur (iii) et également par manque de personnes impliquées.

Tout d'abord une «équipe d'architecture», a été chargée début 2000 de concevoir «Athena», le framework du nouveau code, en contact étroit avec les utilisateurs. Cette équipe a choisi de se baser sur Gaudi [2], le framework créé par LHCb, maintenant développé conjointement par Atlas et LHCb. Depuis 2002, Athena lui même étant devenu relativement stable, l'équipe d'architecture a été étendue pour refléter la diversité des utilisateurs et son rôle est maintenant d'édicter des recommandations sur différents aspects du software d'Atlas. Les différents algorithmes de reconstruction ont été ensuite pour la plupart réécrits en C++, (même si certains ont gardé un module fortran) et de nouveaux algorithmes ont été développés.

En été 2002 a eu lieu la première phase du premier Data Challenge, dont le but était de simuler dans Geant3 des quantités importantes de données avec la description la plus à jour du détecteur, pour permettre la validation des principes du déclenchement. Après une période de mise à jour du logiciel de simulation (qui était «gelé» depuis 1997), le printemps 2002 a été consacré à son débogage et à la mise au point des procédures d'exportation de code et de bookkeeping. La production de simulation proprement dite a eu lieu sur une quarantaine de sites au cours de Juillet-Août 2002. La deuxième phase a été la reconstruction printemps-été 2003 avec le nouveau code C++ avec pour but la production des données nécessaires pour le High Level Trigger TDR et pour un atelier de physique qui s'est tenu à Athènes au printemps 2003.

Une conséquence de la transition progressive vers la reconstruction C++ a été la difficulté d'avoir une base d'utilisateurs suffisamment grande. Le dilemme habituel étant qu'avant de migrer, les physiciens veulent un code garanti sans bogues, ce qui n'est possible que si le code est testé par suffisamment de physiciens dans des configurations suffisamment variées. Pour sortir de cette impasse, et pour encourager la migration de nombreux tutoriaux ont été organisé : au départ assez généraux, ils étaient ensuite plus directement orientés vers la reconstruction d'un détecteur particulier : calorimètre électromagnétique, trajectographe interne ou système muon. Un total de près de 300 personnes ont suivi ces tutoriaux, où sur un ou deux jours, des présentations générales alternaient avec des exercices à faire immédiatement sur ordinateur. 2003 a donc marqué le point où une vaste majorité des physiciens ont renoncé au code fortran grâce :

- à ces tutoriaux
- au succès du Data Challenge
- à l'atelier de physique d'Athènes qui les a motivé
- à l'arrivée du test en faisceau combiné (voir plus loin)

et malgré (il faut bien le dire) :

- une documentation insuffisante
- l'évolution rapide du software
- la réticence à suivre une formation en C++.

En 2003, à l'atelier de physique d'Athènes, plus de la moitié des résultats présentés utilisait le nouveau code, le reste de la communauté a ensuite basculé rapidement. Les performances du code de reconstruction étaient alors similaires au code fortran, avec quelques algorithmes entièrement nouveaux (par exemple l'étalonnage des jets par l'algorithme dit «de H1»), et quelques algorithmes manquant (la reconstruction des conversions de photon).

L'inconvénient de la transition adiabatique est que souvent les algorithmes ont été réécrits en copiant la structure du code fortran et l'emballant dans Athena. En printemps-été 2003 a donc également été mis en place une Reconstruction Task Force [3] dont le but était après le succès de la migration fortran à C++, de définir dans ses grandes lignes les évolutions nécessaires du code de reconstruction avant le début de la prise de données, et d'en faire la conception, sans rentrer dans le détail des algorithmes. Par exemple, il faut découper les algorithmes en modules plus petits (mais pas trop !) et externaliser le maximum de tâches dans des services communs, ce qui permet de faciliter la maintenance, le développement et le perfectionnement des algorithmes. Par exemple un service de description de la longueur de radiation dans le détecteur sera utilisé par les algorithmes de trajectographie (dans le détecteur interne ainsi que dans le spectromètre à muons) et pour l'étalonnage du calorimètre électromagnétique ; il sera également utile pour étudier l'efficacité d'un algorithme de reconstruction des photons convertis. Les recommandations de la RTF (détaillées Section 1.3) ont été bien reçues par la communauté et à Noël 2005 pratiquement tout le code de reconstruction avait évolué en ce sens.

De mai à octobre 2004 a eu lieu une étape très importante dans la préparation d'Atlas avant 2007 : le test en faisceau combiné auquel ont participé tous les détecteurs d'Atlas, baril et bouchon sur deux lignes de faisceau. Le test baril, le plus complet, a mis en jeu les deux technologies de silicium (pixel et bandes), le trajectographe à paille (TRT), un module du calorimètre électromagnétique argon liquide, un module du calorimètre hadronique à tuile, et différents détecteurs de muon. Les détecteurs installés représentaient environ 1% de l'angle solide d'Atlas. Un champ magnétique horizontal courbait les particules chargées comme dans Atlas. Tous les types de particules dans un vaste domaine d'impulsion ont été utilisés, permettant d'étudier sur des données réelles des sujets aussi divers que la correction de Bremsstrahlung pour les électrons, la reconstruction de vertex, l'étalonnage de la reconstruction des jets... Auparavant les données de faisceau test étaient analysées par des logiciels dédiés, qui avaient évolué depuis les premiers R&D du début des années 90, mais pour ce test il a été décidé d'utiliser directement les algorithmes de reconstruction hors-ligne, avec bien sûr une adaptation nécessaire.

Du point de vue software, le but était qu'une large partie du code de reconstruction d'Atlas complet puisse être utilisé tel quel. Cela exige que ce qui était spécifique au test en faisceau (position des détecteurs, asynchronisme du faisceau...) soit très localisé. D'autre part, le traitement des étalonnages et désalignements a dû être implémenté avec des bases de données. L'expérience du traitement de ces données réelles a été très précieuse pour envisager le démarrage du LHC, jusque dans ses aspects sociologiques.

L'année 2004 a été également consacrée au Data Challenge 2 avec un certain nombre d'objectifs, génération des événements avec Geant4 au lieu de Geant3, production avec les outils Grid, reconstruction avec persistance ESD et AOD (voir Section 1.4) des objets dans Pool (le nouveau format de donnée). Du point de vue de la reconstruction, les recommandations de RTF ont été implémentées en ce qui concerne l'Event Data Model (en particulier classes de Trace et de Cellule communes, et code de reconstruction correspondant). La phase de reconstruction s'est terminée au printemps 2005 où les données analysées ont été exposées à l'atelier de physique de Rome en juin 2005.

L'année 2006 a été marquée par la préparation des Calibration Data Challenge (CDC) et System Commissioning (CSC). Ces exercices, préalables au début de la prise de données, mettent autant l'accent sur le réalisme de la simulation et de la reconstruction que sur le volume des données traitées. La simulation a été améliorée pour prendre en compte les «petits» effets, dont une partie a été validée sur les faisceaux tests : détails de la digitisation, description exacte de la matière active ou passive correspondant au détecteur construit. Des désalignements et désétalonnages cohérents avec ceux attendus ont été implémentés. Les générateurs Monte-Carlo ont été mis à jour avec les inputs les plus récents provenant des dernières mesures effectuées au Tevatron (fonction de densité de partons, événement sous-jacent, événements de biais minimum, et avec les générateurs dédiés Next-to-Leading-Order tels que Sherpa ou Alpgen. La production à grande échelle sur la grille a démarré à l'automne 2006.

Coté reconstruction, l'expérience de la reconstruction des faisceaux test a été intégrée, et la simulation du déclenchement de haut niveau fait maintenant partie à part entière du programme de reconstruction : ce qui était indispensable pour que les analyses à venir tiennent compte avec réalisme du déclenchement. Les objets d'analyse (AOD) ont également subi une deuxième itération.

1.1.2 Intégration et tests de validation

Le code de reconstruction est développé en coordination entre de multiples intervenants : les physiciens spécialistes de chaque détecteur concevant la reconstruction spécifique de ce détecteur, les physiciens concevant la reconstruction/identification finale de chaque particule, combinant l'information de la plupart des détecteurs, les physiciens réalisant des études de physique, qui sont les «clients» ultimes, les ingénieurs informaticiens concevant le framework et les outils communs, et bien sur la direction de l'expérience. Ces personnes sont dispersées de la Californie à la Russie, sans qu'il y ait de «noyau dur», cela exige donc beaucoup d'efforts de communication, de planification et d'intégration.

Or, pour diverses raisons, le CERN n'assure plus sa mission traditionnelle de support informatique de l'expérience, si bien que cette mission essentielle doit être distribuée dans les laboratoires. En conséquence, Atlas s'est doté d'un coordinateur de distribution¹. Son rôle est d'établir, étant donné les objectifs généraux, un planning précis et détaillé des développements pour un cycle majeur (environ 6 mois), et de le vérifier au jour le jour. Une «Software Infrastructure Team» réunit entre autres le libraire, le «coordinateur de distribution», l'expert CMT (le logiciel de gestion des logiciels), un libraire d'un site externe pour les questions d'export des logiciels, le responsable des tests, le responsable de l'assurance qualité et le responsable du support utilisateur.

Le nombre important de personnes impliquées et leur dispersion a conduit progressive-

1. En anglais : *release coordinator*

ment à un modèle de développement où des outils d'aide à l'intégration ont été développés. Le cycle de développement s'est stabilisé avec environ une distribution de production tous les 6 mois (numérotée $n.0.0$), suivie éventuellement d'une branche de distributions de débogage numérotées $n.0.i$. Après une distribution de production n a lieu des distributions de développement $n.j.0$ environ toutes les trois semaines, qui aboutiront six mois plus tard à la distribution de production $(n + 1).0.0$. Les distributions de production doivent être complètement fonctionnelles (même si il est arrivé que la partie simulation soit découplée de la partie reconstruction), et les physiciens sont encouragés à migrer vers une nouvelle distribution de production quand elle est disponible. Par contre une distribution de développement peut ne pas être complètement fonctionnelle, et seules les personnes contribuant directement aux développements du code les utilisent.

L'intégralité du code d'Atlas est sous CVS, divisé en près d'un millier de packages. En général un package correspond à une unité de code avec une fonctionnalité bien définie, et est compilée sous forme de librairie partagée ; cette librairie peut être chargée et exécutée seule, ou du moins avec le minimum d'autres librairies nécessaires, ou bien faire partie d'une grande application comme la reconstruction ou la simulation. Certains packages sont des packages de configuration, comportant des scripts Python qui permettent de charger les librairies partagées et de configurer le code correspondant pour une application donnée, d'autres enfin définissent l'interface avec les logiciels externes. Le package est l'unité de base de développement : une distribution est entièrement définie par la liste de tous les packages et de leurs tags CVS. Les packages sont organisés dans une arborescence CVS (analogue à une arborescence de répertoires Unix) : en tête de l'arborescence, des répertoires tels que InnerDetector ou LArCalorimeter groupant l'intégralité du code concernant spécifiquement le détecteur interne ou le calorimètre, des répertoires comme Tracking (code commun au détecteur interne et au spectromètre à muon) et Calorimeter (pour les calorimètres liquide argon ou à tuile), le répertoire Reconstruction groupant soit des utilitaires communs, soit des algorithmes de reconstruction combinés (comme les jets ou l'identification des particules). Les packages sont gérés par CMT qui permet de définir quelle action est nécessaire sur chaque package, ainsi que les dépendances entre packages.

Pour maximiser la qualité des distributions, un modèle d'intégration «au fil de l'eau» a été mis au point progressivement pour permettre à chaque développeur de pouvoir développer son propre code (package) contre les package les plus récents fournis par les autres développeurs. Ce modèle repose d'une part sur TagCollector (développé au LPSC Grenoble) pour collecter les package/tag d'une distribution donnée, et d'autre part sur le trio ATNIGHT/NICOS/RTT (les deux premiers développés à Brookhaven et le troisième à UCL London) qui permettent de construire et tester chaque nuit une nouvelle distribution ².

TagCollector est une interface web devant une base de données relationnelle AMI [4], qui sert également à la gestion des ensembles de données mais cet aspect ne sera pas abordé). Tous les packages/tags d'une distribution sont recensés dans la base de données, y compris ceux de la distribution future. Chaque package a une poignée de responsables (qui contribuent directement au code) ; seul un de ces responsables (ou un responsable du répertoire correspondant) peut changer la version d'un package dans la distribution future. TagCollector peut gérer plusieurs futures distributions en même temps : la distribution de développement, une distribution de débogage de la distribution de production précédente, voire une distribution spéciale pour par exemple une activité de commissioning.

Tous les soirs à 9h, ATNIGHT se déclenche, relève la liste des package/tag de la distri-

2. En anglais : *nightly release*

bution future, importe³ (c.à.d extrait de CVS) les packages modifiés depuis la veille, recompile tous les packages qui doivent l'être (c.à.d tous les packages modifiés mais aussi ceux qui dépendent, parfois en cascade, des packages modifiés), puis copie sous afs la distribution complète ainsi obtenue, et finalement lance NICOS qui effectue les tests automatiques courts (décrit plus loin). Si bien que le lendemain matin, la distribution intégrant tous les développements de la veille est disponible. Puis le système RTT de test se déclenche : ces tests en batch sont disponibles dans la soirée ou le lendemain.

NICOS et RTT utilisent des tests qui sont définis en XML : quel job exécuter, sur quelles données, suivi par quelles actions. Les tests de NICOS sont des tests prévus pour tourner interactivement sur quelques événements : ils permettent de tester l'intégration des nouveaux packages, vérifier l'absence de plantage et de messages d'erreur, de tester leur utilisation mémoire et CPU, et ce dans de multiples configurations. Les tests RTT sont lancés en batch et portent sur plusieurs centaines à milliers d'événements, ils permettent ainsi la production d'histogramme de référence pour tester la performance des algorithmes.

Pour apprécier l'importance de ce système, prenons le scénario d'un physicien qui développe un algorithme particulier d'identification des électrons. Au départ, il travaille contre la dernière distribution de production, par exemple 12.0.6. Puis, il veut utiliser un nouveau code de tracking qui a été introduit dans la dernière distribution de développement, il va donc travailler contre cette distribution 12.1.0. Puis, il s'avère que ce nouveau code était erroné, et a été corrigé depuis dans les distributions de nuit, il va donc utiliser les distributions de nuit. Au fur et à mesure de ses propres développements, il met à jour lui-même la version de son package dans TagCollector, qui est ensuite intégré à la distribution de nuit. Il peut ainsi indiquer à un collègue qui souhaite utiliser son outil quelle distribution de nuit utiliser. En l'absence de distribution de nuit, il aurait été obligé de faire lui-même l'importation et la compilation des possibles multiples packages nécessaires au nouveau code de tracking, avec les risques d'erreur que cela comporte, surtout si ces packages sont mis à jour régulièrement. D'autre part, il aurait probablement négligé de tester son code dans toutes les configurations possibles, et des problèmes ne seraient apparus qu'au moment de finaliser une distribution.

Prenons le scénario d'un autre physicien qui travaille sur un code de calorimétrie largement utilisé. Sa problématique est différente : il pourrait travailler et tester son code sur une distribution de production, mais dans ce cas il ne testerait pas l'impact de ses modifications sur les algorithmes qui dépendent du sien (ses «codes clients »). En mettant ses changements au fur et à mesure dans la distribution de nuit, il peut immédiatement voir les impacts de ses changements sur ses codes clients : erreur de compilation, plantage ou messages d'erreur à l'exécution sur quelques événements, ou performance affectées. Les codes clients pourront être corrigés dès la distribution de nuit suivante. Bien sur, il aura essayé de minimiser l'impact sur ses clients, voire les aura prévenus si une modification simultanée était nécessaire (TagCollector permet même d'envoyer un message à tous les responsables des packages clients d'un package donné). En l'absence de distribution de nuit, ces problèmes auraient été découverts au moment de la finalisation de la distribution, et auraient probablement exigé plusieurs itérations.

Clairement le système ne fonctionne que si les développeurs sont disciplinés. Si un développeur d'un algorithme de tracking central fait une erreur et mets dans la distribution de nuit un package qui ne marche pas, il affecte tous ceux qui travaillent sur le tracking, puis en cascade sur l'identification des particules, puis sur le développement de l'analyse. Seul, les développements sur le calorimètre ne sont pas affectés. Mais plus le package est en aval, proche de l'analyse,

3. en anglais *check-out*

plus il dépend en cascade d'un très grand nombre de package, donc plus il est probable que ce package ne marche pas dans les distribution de nuit. La responsabilité est donc d'autant plus grande que le package est en amont.

Pourtant, bizarrement, le système fonctionne relativement bien, le code d'analyse (exigeant toute la chaîne de reconstruction en amont) fonctionne une fois sur deux, en intégrant les erreurs dans les dernières modifications et les problèmes d'infrastructure comme les hoquets d'afs. Les physiciens sont habitués au système et comprennent l'intérêt de la réussite des distributions de nuit, et donc sont prudents dans leur soumission de tags. De plus les changements majeurs d'infrastructure (changement de compilateur, changement majeur de package de base comme Gaudi ou LCG), généralement perturbateurs, sont confinés aux premières distribution de développement, de telle sorte que ces développements ne soient plus dangereux quand s'approche la distribution de production.

Par contre, une difficulté bien identifiée persiste dans Atlas : des développements nouveaux sont délibérément réalisés dans la distribution de débogage ou bien (ce qui revient au même), la date d'une distribution de production est maintenue alors qu'une partie des développements prévus n'est pas terminée. Du coup, les tests de performance réalisés avant la fermeture d'une release sont souvent insuffisants. Il s'ensuit une longue période de fin de développement et débogage «sur la branche» (dans les distribution $n.0.x$), alors que de nouveaux développements ont lieu simultanément «sur l'arbre» (dans les distribution $n.x.0$).

Il a été mentionné que les outils TagCollector/ATNIGHT pouvaient gérer plusieurs distributions en parallèle, mais malheureusement l'homme, même physicien, n'a qu'un seul cerveau. Le développeur se forge en mémoire une représentation des changements récents de son propre code, des codes dont il dépend, et du code de ses clients, mais il ne peut pas garder en permanence deux versions de cette représentation.

Autrement dit on ne peut pas demander à une personne, ni à une communauté, de travailler simultanément à des nouveaux développements «sur l'arbre» tout en finalisant les développements «sur la branche». Ce qui se passe en pratique est que la communauté est focalisée «sur la branche», laissant «l'arbre» en déshérence, alors même que certains développeurs qui ne sont pas impliqués dans les développements récents ou bien qui les avaient validés à temps aimeraient bien pouvoir travailler sur l'arbre. D'où une perte de temps et une inefficacité certaine. Il y a deux façons de sortir de cette situation : retarder la distribution si la date n'est pas contrainte de façon externe, ou bien, et ce sera le cas dans le futur immédiat avec le démarrage du LHC, limiter les développements à ce qui est vraiment nécessaire.

1.1.3 *l'eXtreme Programming*

Le style de développement dans Atlas s'est forgé petit à petit à partir des influences d'un certain nombre de personnes, expert programmeurs ou physiciens pragmatiques, après moult essais et repentirs. La transition C++ a été assez lente, de 1999 à 2006, si bien que plusieurs modèles ont été essayé. Les sections précédentes décrivent le modèle actuel, il est intéressant de le comparer à une méthode de développement populaire récemment chez les informaticiens, l'eXtreme Programming. L'eXtreme Programming est une méthode de développement souple et pragmatique datant du début des années 2000, en réaction à des méthodes beaucoup plus formelles, telles que USDP (Unified Software Development Process).

USDP définit quatre étapes : inception (ou collecte de l'ensemble des informations), la conception, l'implémentation et le déploiement, ce qui ne correspond pas au nombreuses

itération du développement d'un code de reconstruction. Ces étapes sont formalisées et ponctuées par la rédaction d'un document des requis⁴ et cas d'usage⁵, une revue de ce document, la rédaction d'un document de conception, une revue de ce document, pour finir par l'implémentation et la revue du code. L'utilisation de cette méthode dans Atlas n'a pas abouti, en raison principalement de sa lourdeur. Il en est resté par contre des bonnes pratiques, comme de discuter en terme de cas d'usage et donc d'interface avant de se lancer dans l'implémentation. Les revues existent toujours mais sont ponctuelles et sur un domaine limité. Pour résoudre des problèmes précis des task force sont mises en place, chacune rédigeant un document avec des recommandations circonstanciées, avant de s'auto-dissoudre (se pose alors parfois le problème de l'implémentation de ces recommandations).

L'eXtreme Programming [5] repose sur 13 principes, nous allons voir dans quelle mesure la méthode de développement empirique adoptée par Atlas s'en rapproche ; un score sur 5 a été donné évaluant la mesure avec laquelle la pratique dans Atlas se rapproche de eXtreme Programming.

- Pratiques de programmation

- Conception simple, pas de développement «génériques» qui ne serviront jamais que dans un cas particulier : 4/5 *C'est en général le cas dans Atlas. La plupart des développeurs sont des physiciens pour qui la programmation n'est pas un but en soi.*
- Remaniement, pas d'hésitation à réécrire un code devenu trop complexe ou inadapté 3/5 *C'est en général le cas dans Atlas, même si ce n'est pas forcément spontané, on n'aime pas toucher à ce qui marche. Néanmoins, l'EDM par exemple a été réitéré plusieurs fois. D'autre part, un mini cycle de développement courant est : un algorithme de reconstruction est écrit par un développeur A, qui réalise qu'il a besoin de nouveaux outils ou bien d'extension d'outils existant, l'algorithme de reconstruction est néanmoins développé sans ces développements. Puis les outils ou extension d'outils sont réalisés par ce développeur ou par un autre. Finalement, l'algorithme en question est simplifié en utilisant les nouveaux développements.*
- Développement piloté par les tests unitaires, le test d'un élément de code est développé simultanément voire avant le code en question 1/5 *Seuls les développeurs Core ont cette pratique, qui serait pourtant largement généralisable.*
- Tests de recette aussi appelé test de régression, la fonctionnalité est testée rapidement et automatiquement, les tests restent en place pour vérifier que le code ne régresse pas suite à de nouveaux développements 3/5 *Les tests automatiques des algorithmes de reconstruction se sont mis en place au cours de l'année 2006. Il y a encore beaucoup à faire mais l'intérêt n'en fait aucun doute pour personne. Cette activité est reliée au monitoring des données : le monitoring évalue en permanence des données différentes avec un software stable, alors que les tests de recette évaluent en permanence des données identiques avec un software qui évolue.*

- Pratiques de collaboration

- Programmation en binôme, les développeurs travaillent à deux devant le même écran 0/5 *pas appliqué du tout et difficilement applicable.*
- Responsabilité collective, tout développeur peut intervenir sur toute partie du code 1/5 *de 1 à 5 personnes peuvent travailler simultanément sur chaque package, mais*

4. en anglais *requirements*

5. en anglais *use cases*

étant donné la taille et la complexité du code il n'est pas envisageable que tout le monde intervienne sur n'importe quelle partie du code.

- Règles de codage 4/5 *des règles de codage existent effectivement dans Atlas [6], elles sont relativement bien suivies.*
- Métaphore, des métaphores sont utilisées pour décrire le fonctionnement du code 0/5 *difficilement applicable, à moins qu'on considère la reconstruction d'objets comme une métaphore à part entière.*
- Intégration continue 5/5 *C'est le système de distribution de nuit décrit dans la section précédente.*
- Pratiques de gestion de projet
 - Livraisons fréquents 5/5 *C'est le système de distribution de développeurs toutes les trois semaines, avec les distribution de production tout les 6 mois.*
 - Planification itérative 5/5 *C'est même un peu trop le cas dans Atlas.*
 - Client sur site 3/5 *Le client étant les physiciens, et les développeurs étant physiciens eux-même, il y a un couplage naturel assez fort, mais qui pourrait être amélioré. De plus le décalage horaire à un effet néfaste certain.*
 - Rythme durable *sans commentaire.*

L'eXtreme Programming a été décrit pour typiquement une équipe d'une dizaine de personnes travaillant sur un même site, alors que le développement du software dans Atlas implique environ 200 personnes dispersées. Pourtant, les pratiques dans Atlas se rapprochent en général assez bien de l'eXtreme Programming. Les secteurs où il serait bénéfique de faire des progrès dans Atlas sont :

- les tests de recette : il y en a jamais assez.
- les test unitaires : une pratique peu développée et qui gagnerait à l'être.
- client sur site : il faut lutter en permanence contre la tendance de séparation entre «ceux qui écrivent du code» et «ceux qui font des plots» .
- la planification : éviter les glissement perpétuels.
- remaniement : devrait être fait plus systématiquement.

1.2 Architecture

1.2.1 Orienté objet, mais pas trop

La principale décision d'architecture à la base de Gaudi/Athena est la séparation entre les algorithmes et les données. Cette décision peut paraître paradoxale pour un expert C++/orientation objet car justement le principe de base de la programmation objet est de grouper données et algorithmes dans un même objet. En effet, les premiers essais de transposition d'algorithme fortran en code C++ complètement orienté objet ont été des échecs, principalement pour deux raisons : les performances (en terme d'efficacité) obtenues n'étaient pas suffisantes, le code résultant était très complexe (le départ de l'auteur principal entraînant une réécriture de zéro), les dépendances dans le code étaient inextricables. Même des versions plus légères, où par exemple une trace pouvait elle-même réajuster ses paramètres lorsqu'on lui ajoutait ou supprimait un impact, ont du être abandonnées, à nouveau pour des raisons de complexité et de dépendance. La transition Fortran-C++ décrite en début de chapitre s'étant déroulée sur plusieurs années, par comparaison à d'autres expériences telles BaBar ou CDF/D0 où cette transition a été plus rapide avec une contrainte de temps plus tendue, cela a permis en plusieurs itérations la mise en place d'une approche «orientée objet mais pas trop» . En effet, les difficultés que posent l'approche orientée objet «pure et dure » , sont également connues dans le monde des informaticiens [7].

L'approche orientée objet «pure et dure » demande la définition d'objets simples, en précisant même : avec 7 ± 2 méthodes dans l'interface car ce serait le nombre d'éléments qu'un cerveau humain peut appréhender simultanément. Il ne doit pas y avoir d'objet manager, d'objet dieu : le comportement complexe de l'algorithme viendra des interactions des objets simples. Or l'émergence de la complexité, c.à.d comment des règles ou comportements simples et locaux donnent lieu à des comportements globaux complexes, est une science à part entière, en plein développement.

Un domaine d'étude très actif est l'étude des insectes sociaux [8] : comment le comportement collectif apparemment parfaitement réglé de la fourmilière ou de la ruche émerge du comportement des fourmis ou des abeilles, dont chacune a un cerveau de un millimètre cube. Un exemple récent de ces études est la modélisation de la dynamique de cimetières de fourmis : il est connu depuis fort longtemps que de nombreuses espèces de fourmis ont pour habitude d'entasser leur cadavres (ou leur couvain) en tas (soit un processus de clusterisation). Dans une série d'expériences, des cadavres de fourmis ont été régulièrement répartis dans une arène circulaire, à laquelle a accès une colonie de fourmis. Après quelques heures des clusters stables apparaissent (voir Figure 1.1). Ce comportement peut être entièrement expliqué (c.à.d la taille caractéristique des clusters et la distance entre clusters peut être retrouvée par le calcul ou par simulations) en supposant qu'une fourmi, se déplaçant plus ou moins aléatoirement, a une probabilité de saisir un cadavre d'autant plus grande que la densité locale de cadavres est faible, et une probabilité de le redéposer d'autant plus grande que la densité locale de cadavres est grande.

Mais, il n'y a aucune raison de penser que la reconstruction des particules dans un calorimètre ou un trajectographe puisse être exprimable à partir d'interactions d'objets simples, ni d'espérer qu'un tel algorithme fonctionne mieux que des algorithmes plus classiques. En reprenant, l'exemple de la Figure 1.1 : comment modifier le comportement des fourmis pour qu'elles forment des clusters de forme caractéristiques des dépôts d'énergie d'électrons, par exemple plus long dans une direction donnée ? Tant bien même un tel algorithme existerait : supposons qu'une étude des performances montre une déficience, par exemple pour les particules de faible impulsion. Les physiciens concepteurs de l'algorithme devraient alors résoudre deux problèmes :

- comprendre la raison de la déficience, sachant que le fonctionnement même de l'algorithme est par définition peu intuitif
- trouver quelles modifications des objets simples, ou quel nouvel objet simple introduire pour améliorer la déficience identifiée, sans perturber par ailleurs le fonctionnement de l'algorithme

1.2.2 Séparation données-algorithmes

L'approche orientée objet «pure et dure» a donc été abandonnée pour une approche plus pragmatique dont la séparation entre les algorithmes et les données est la pierre angulaire. Donc, ont été séparés d'un côté le modèle de donnée (EDM pour Event Data Model), de l'autre côté les Algorithmes, les Tools et les Service (la majuscule indique que ce sont des classes d'Athena). Figure 1.2 illustre le fonctionnement en tableau noir : le flux de données apparent est que le tracking lit les digits du trajectographe et fournit des traces à l'identification électron/photon, le clustering dans le calorimètre lit les CaloCells et les fournit à l'identification électron/photon, l'identification électron/photon fournit électrons et photons. En pratique, chaque algorithme lit ses données d'entrée et écrit ses données de sortie dans le Transient Event Store (TES), géré par

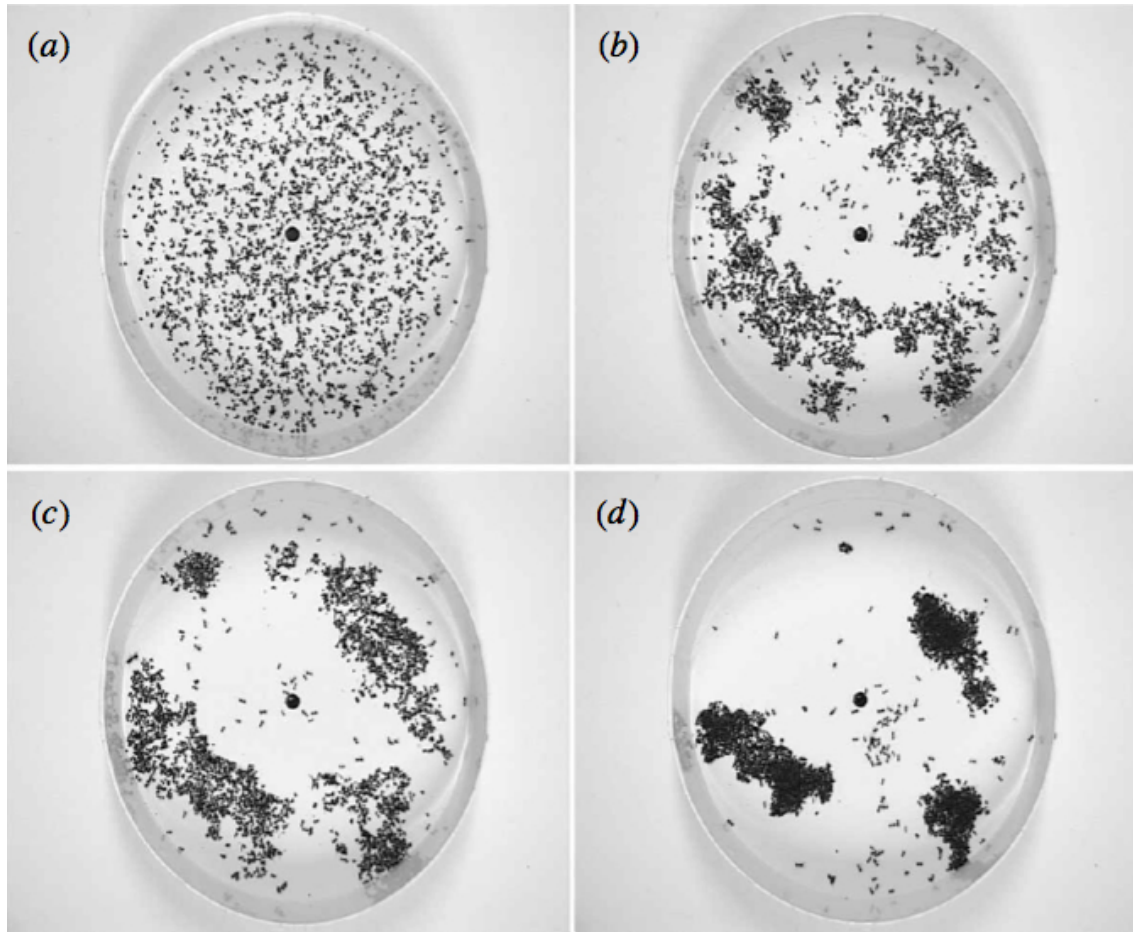


FIGURE 1.1: Dynamique d'agrégation (ou «clusterisation») de cadavres de fourmis. Images réelles de l'arène à $t=0$ (a), $t=3h$ (b), $t=6h$ (c), $t=36h$ (d). [8]

StoreGate (en pratique, seuls les pointeurs sont manipulés). A chaque objet est associée une clé (chaîne de caractère) telle que le couple type d'objet/clé soit unique. La définition de l'Event Data Model est la définition de tous les objets destinés à être écrits dans le Transient Event Store, une grande partie d'entre eux étant également destinés à être persistifiés, c.à.d écrit sur fichier. Le flux entre les fichiers et le Transient Event Store sont gérés par les convertisseurs, chaque convertisseur étant responsable de l'écriture et la lecture d'un type d'objet (au sens C++). Le modèle en tableau noir avec un Event Data Model servant de moyen de communication entre algorithmes évite la dépendance directe entre algorithmes et favorise l'interchangeabilité de ces algorithmes. De plus, la gestion de la mémoire, qui est une difficulté connue de C++ (puisque tout objet explicitement créé doit être explicitement détruit), est simplifiée car tout objet écrit dans le Transient Event Store est automatiquement détruit à la fin du traitement de l'événement.

Les classes de l'EDM vont correspondre aux objets reconstruits (trace, cluster, jet, etc...) et leur rôle est de contenir l'information. Leur «intelligence» est limitée à présenter cette information de la façon la plus efficace pour les différents clients. Par exemple une `CaloCell`, cellule du calorimètre, peut aussi bien donner sa direction en coordonnées cartésiennes qu'en η - ϕ , mais collecter les cellules du calorimètre dans une direction donnée sera l'oeuvre d'un Tool séparé.

L'Algorithme est l'unité de traitement des données de base. Un Algorithme lit typiquement un (ou parfois plusieurs) objet, et en écrit un (ou rarement plusieurs), par exemple un

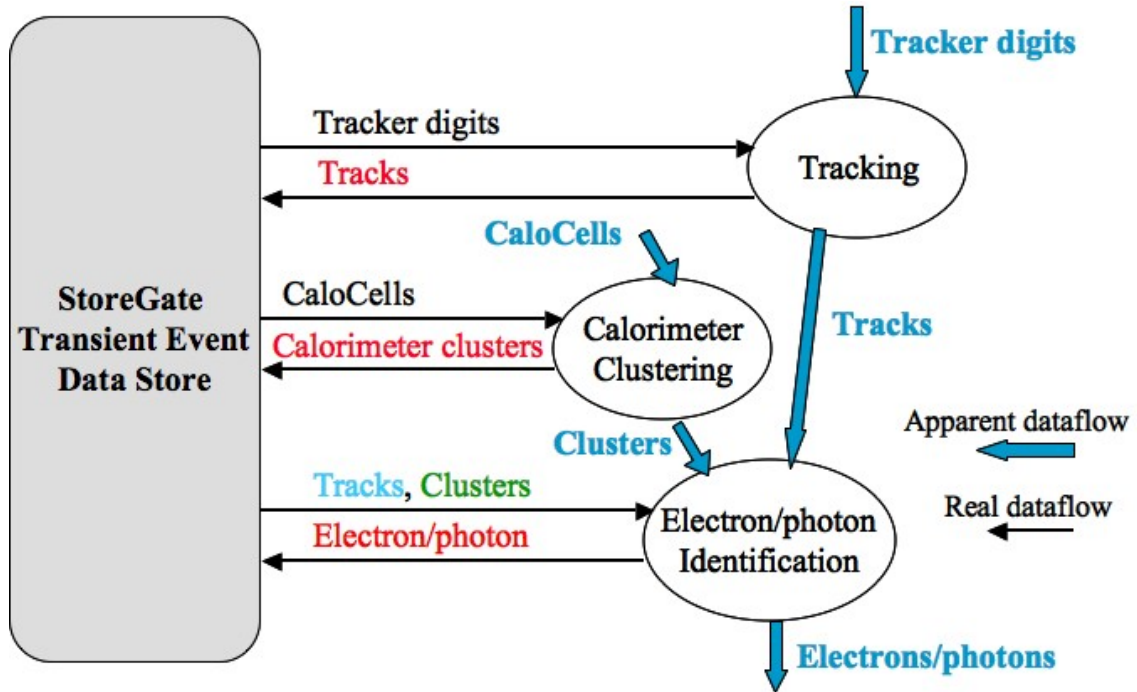


FIGURE 1.2: Illustration du fonctionnement en tableau noir d'Athena.

Algorithme lit les CaloCell pour écrire les CaloTower, un autre lit les CaloTower pour écrire les CaloCluster, un autre lit les CaloCluster et les TrackParticle pour écrire les egamma (objet contenant toute l'information nécessaire à l'identification des électrons ou photons), un dernier lit les egamma pour écrire les Photon et les Electron.

1.2.3 Event Data Model de la simulation et des données brutes

L'intégralité des étapes de la simulation est présentée sur la Figure 1.3, qui montre le niveau de détail extrême avec laquelle la simulation est faite. Toutes les étapes ont lieu dans Athena. Le point clé en est les Raw Data Object (RDO) qui sont les objets C++ à l'entrée de la reconstruction, et qui sont les données persistantes pour le Monte-Carlo. Le signal électronique au niveau de l'électronique Front-End subi un dernier traitement de réduction par les Read-Out-Driver (ROD) avant d'être stockés sous forme de flux de bits (ByteStream), et converti en RDO par le ByteStreamConversionSvc dès le début de la reconstruction des données. Il peut paraître choquant que le format de données persistantes ne soit pas le même pour données réelles et simulation Monte-Carlo, dans la mesure où tous les efforts sont faits pour que la simulation Monte-Carlo reproduisent les données aussi fidèlement que possible :

- La raison pour laquelle les données Monte-Carlo ne sont pas stockés sous forme de ByteStream est que le ByteStream n'est pas adapté à sauvegarder la vérité Monte-Carlo.
- La raison pour laquelle les données réelles d'Atlas ne sont pas stockées sous forme de RDO est pour une question de robustesse : le passage de ByteStream à RDO implique un décodage en-ligne non trivial où le moindre bogue (dans le software comme dans l'électronique front-end ou les ROD) peut donner lieu à des pertes d'information catastrophiques. La persistance du ByteStream permet de corriger ces problèmes éventuels après coup lors du décodage vers les RDO.

- A titre de test, les RDO (en particulier les RDO Monte-Carlo) peuvent être convertis en ByteStream et inversement.

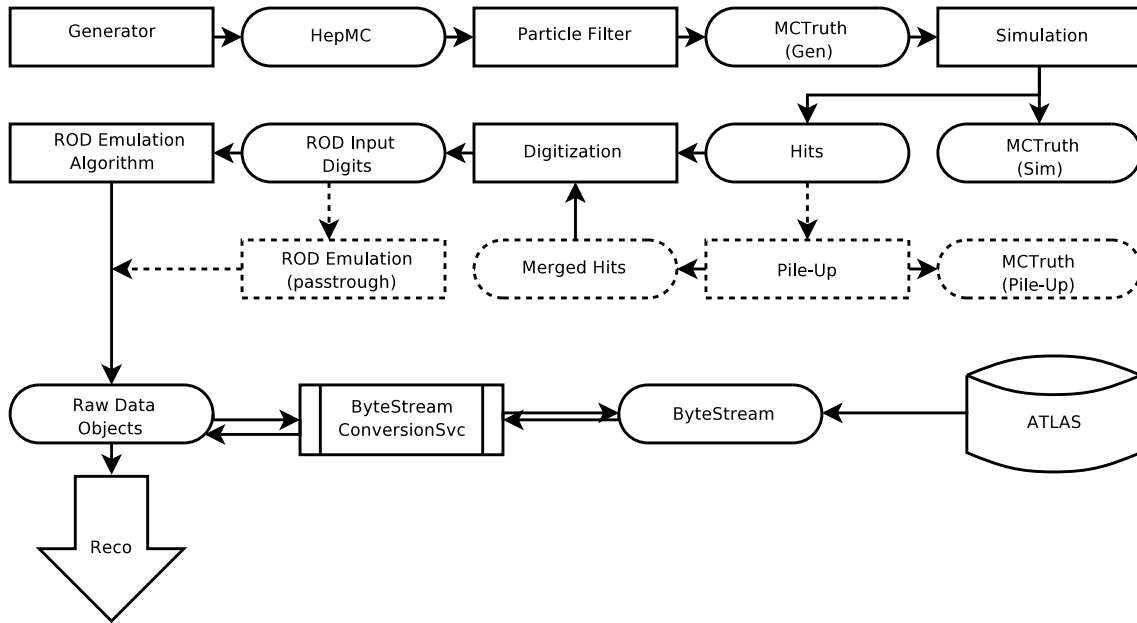


FIGURE 1.3: Event Data Model de la simulation et des données brutes.

La simulation Monte-Carlo commence avec les générateurs : la plupart des générateurs (Pythia, Herwig, etc...) ont été interfacés dans Athena pour écrire l'ensemble des particules sous forme HepMC, un format C++ persistant qui permet sauvegarder en plus des particules les relations mères filles, voire les polarisations. La simulation du détecteur commence avec un filtre qui ne garde que les particules primaires susceptibles de laisser un signal dans le détecteur. Puis Geant4 procède à la simulation proprement dite dont la sortie est d'une part les Hits, c.à.d les dépôts d'énergie dans toutes les parties sensibles du détecteur, d'autre part toutes les particules secondaires. La sortie de la simulation comprend les Hits et les particules générateur, primaires et secondaires dans le même format HepMC qui permet une continuité des relations mère/fille entre les particules secondaires, les particules primaires, jusqu'aux particules de l'interaction primaire. La digitisation lit les Hits en déduit les signaux au niveau de l'électronique Front-End (ROD Input Digits), et simule le traitement opéré dans les ROD pour écrire finalement les RDO, simultanément aux particules vraies. Optionnellement, l'étape de simulation des ROD peut être omise pour étudier le détail du fonctionnement des ROD.

La simulation de l'empilement se fait au niveau de la digitisation : les Hits de l'événement de physique sont mélangés aux Hits des événements de biais minimum, en tenant compte des événements des collisions suivantes et précédentes, en fonction de la durée de la mise en forme qui varie suivant les détecteurs.

1.3 Structure des algorithmes de reconstruction

Cette section donne quelques points clés sur la structuration des algorithmes de reconstruction tirés en grande partie du rapport de la Reconstruction Task Force (RTF) [3] dont il a déjà été question, avec quelques mises à jour. Le contenu algorithmique de ces algorithmes sera abordé dans le chapitre 2.

1.3.1 Survol

La reconstruction complète d'un événement s'opère en trois grandes étapes. Dans une première étape, la reconstruction s'opère séparément dans chaque détecteur et permet d'obtenir, d'un côté les cellules, ou bien les clusters qui sont des groupes de cellules géographiquement proches, de l'autre les traces dans le trajectographe interne et dans le spectrographe à muon. La deuxième étape est la reconstruction combinée (voir Figure 1.4) où les informations des détecteurs sont combinées pour arriver à une meilleure estimation et au calcul de variables d'identification de particules. La troisième d'étape est la préparation de l'analyse (voir Figure 1.5), ou essentiellement les variables calculées au cours de l'étape précédentes sont exploitées pour arriver à des objets correspondants aux particules nécessaires à l'analyse : électron, photon, muon, jet (avec étiquetage des b), τ et énergie transverse manquante.

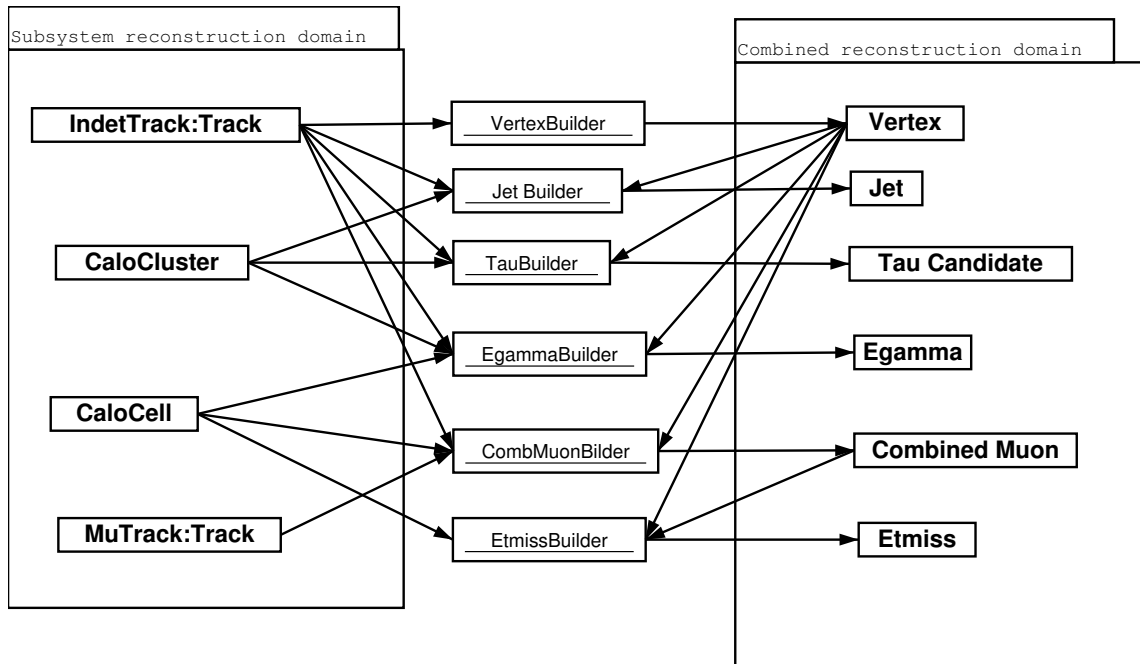


FIGURE 1.4: Flux de données pour la reconstruction combinée [3].

Même si les flux de données représentés sur les Figures 1.4 et 1.5, sont simplifiés, il faut insister sur le fait qu'il n'y a jamais de boucles ni d'itérations (le flux de donnée est un graphe acyclique) ni de modification en continu d'un objet (sauf à l'intérieur d'un algorithme). Dès qu'un objet est écrit dans le Transient Event Store il n'est plus modifiable par un autre algorithme (sauf dans quelques cas très localisés et contrôlés). S'il s'avère nécessaire, par exemple, de refaire le tracking d'une trace sous l'hypothèse électron, la trace originale ne sera pas modifiée, ce sera fait dans un autre algorithme en aval sur une copie de la trace. Il est donc nécessaire de pouvoir partager du code entre algorithmes, ce qui est décrit dans la section suivante.

1.3.2 Des Algorithmes aux Tools

Un Algorithme est exécuté une et une seule fois pour chaque événement⁶, et n'a pas d'interface propre. En pratique, le code de l'Algorithme est relativement simple, un Algorithme

6. Pour être tout à fait exact, plusieurs instances d'un même algorithme (mais configurées différemment) peuvent chacune être exécutée une et une seule fois pour chaque événement.

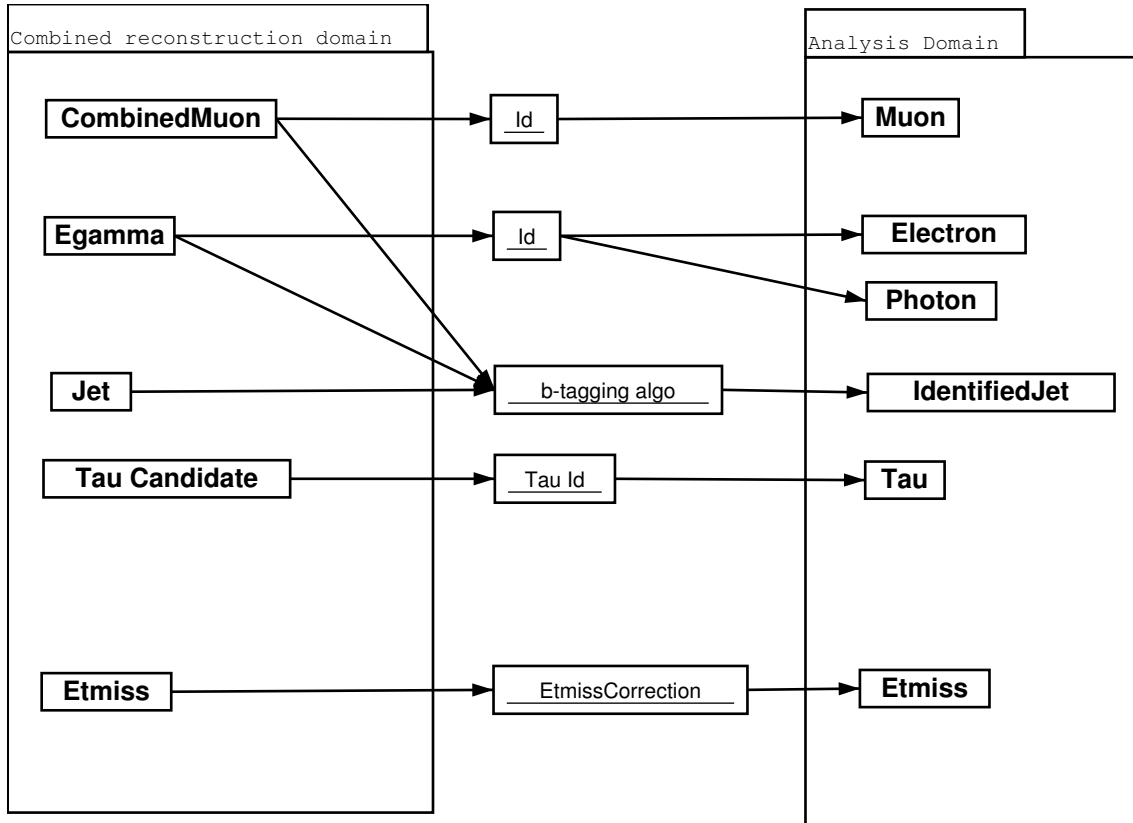


FIGURE 1.5: Flux de données pour la préparation des objets d'analyse (AOD) [3].

exécutant un certain nombre de Tools. Un Tool est une unité de code modulaire, d'interface dédiée, qui peut être exécutée à volonté par des Algorithmes différents. Il peut y avoir dans un même job plusieurs instances du même Tool (configurées différemment) exécutées par plusieurs Algorithmes, ou bien plusieurs Algorithmes peuvent partager la même instance de Tool. Un Tool va par exemple extrapoler une trace ou bien calibrer une collection de cluster. Le Tool permet de partager le même code entre Algorithmes.

Un Service est similaire à un Tool, sauf qu'il ne peut y avoir qu'une seule instance d'un service donné dans un job, qui peut être utilisée par n'importe quel Algorithme ou Tool. Des exemples de Service sont : le Service de champ magnétique, pouvant donner le champ magnétique en tout point du détecteur, le Service de géométrie qui crée la géométrie de tout le détecteur, créant des objets spécifiques successibles de donner les positions correspondant à chacun des canaux électroniques.

L'utilisation massive de Tools proposée par la RTF est indispensable pour le déclenchement de haut niveau. L'«Event Filter» (correspondant au troisième niveau) réutilise le code de reconstruction hors-ligne, mais avec des impératifs de rapidité au moins 10 fois plus élevés (1 s par événement). Pour y parvenir, une partie de la reconstruction n'est faite que dans des régions d'intérêt (ROI). Techniquement, c'est le même Tool qui va être exécuté pour chaque ROI, alors qu'hors-ligne il sera exécuté sur tout l'événement. Cela permet donc le partage effectif de software entre hors-ligne et Event Filter.

1.3.2.1 Un Tool particulier : CaloNoiseTool

A titre d'exemple, et parce que ce Tool a été développé au LAL [9], le Tool CaloNoiseTool est décrit sommairement dans cette section. Le but de ce Tool est de pouvoir fournir pour toutes les cellules du calorimètres une estimation du bruit attendu : bruit électronique, bruit d'empilement ou bruit total. Il est utilisé pour pouvoir comparer le signal du calorimètre à son bruit attendu, ce qui est extrêmement utile pour les calculs d'énergie manquante (voir Section 2.9), de jets (Section 2.8.1), et est également utilisé pour l'identification des τ ou des muons. La dérivation du bruit est schématisé Figure 1.6. Les étapes du calcul du bruit sont les suivantes :

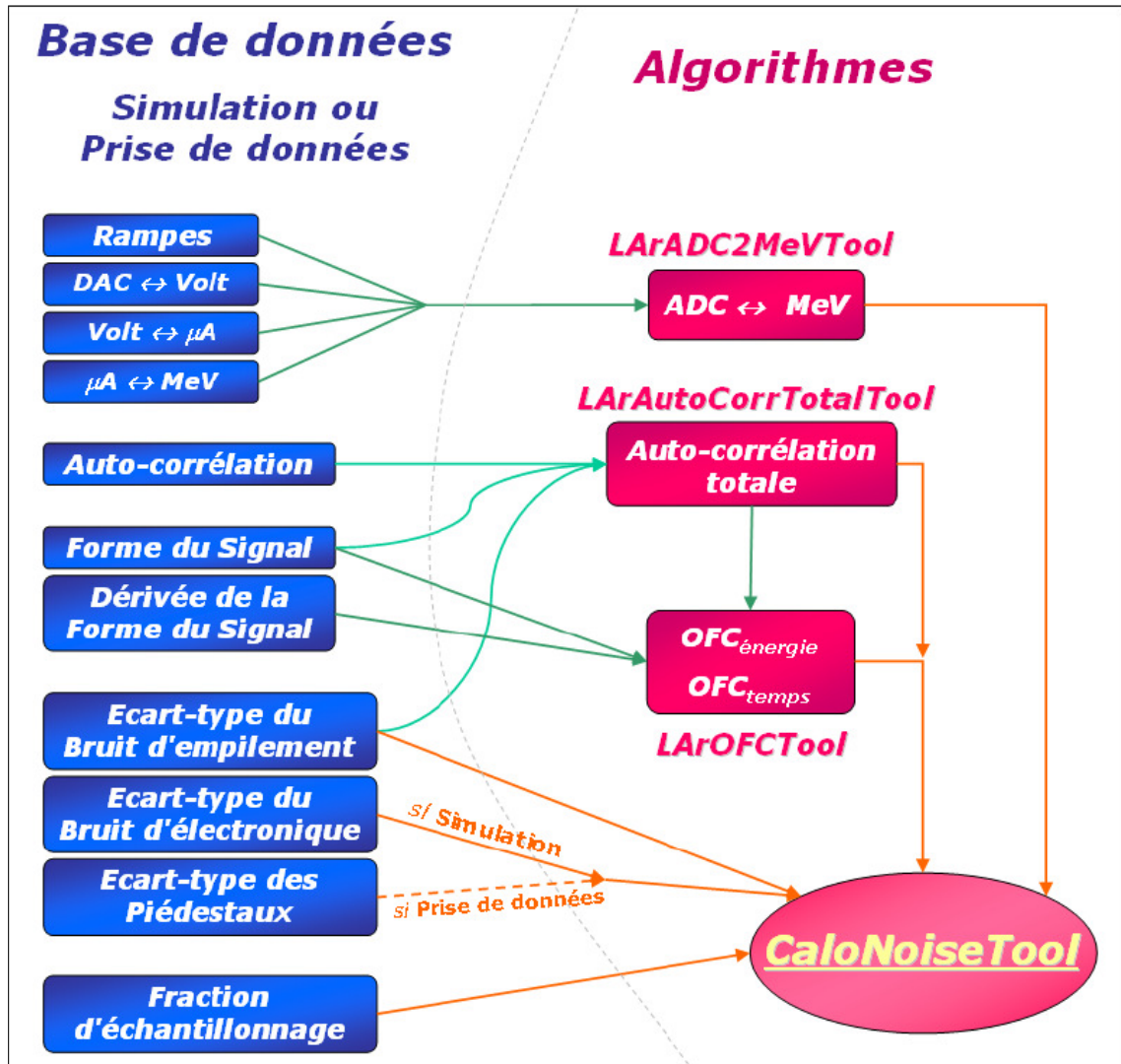


FIGURE 1.6: Dépendances du Tool CaloNoiseTool. A gauche, les données stockées dans la base de données, au milieu les Tool effectuant les calculs intermédiaires d'étalonnage [9].

- Le calcul du facteur de conversion de coups d'ADC en énergie à l'échelle électromagnétique, est effectué par un Tool séparé LArADC2MeVTool. Les ingrédients nécessaires sont stockés dans une base de données. Certains ingrédients sont immuables et ne dépendent que de la région du calorimètre, d'autres (comme les rampes qui sont les pentes issues de la procédure d'étalonnage par injection contrôlée de signal) varient cellule par cellule et peuvent également évoluer au cours du temps.

- La matrice d'auto-corrélation du bruit est calculée par un Tool séparé LArAutoCorrTool. Le calcul nécessite de connaître les corrélations entre les différents échantillons temporels (pour le bruit électronique comme pour l'empilement) et la forme du signal.
- Les coefficients de filtre optimal («Optimal Filtering Coefficients ») sont calculés à partir de la matrice d'auto-corrélation et la forme du signal.
- CaloNoiseTool effectue le calcul final. Un diagramme parmi ceux pouvant être obtenus à partir des sorties de CaloNoiseTool est visible sur la Figure 1.7.

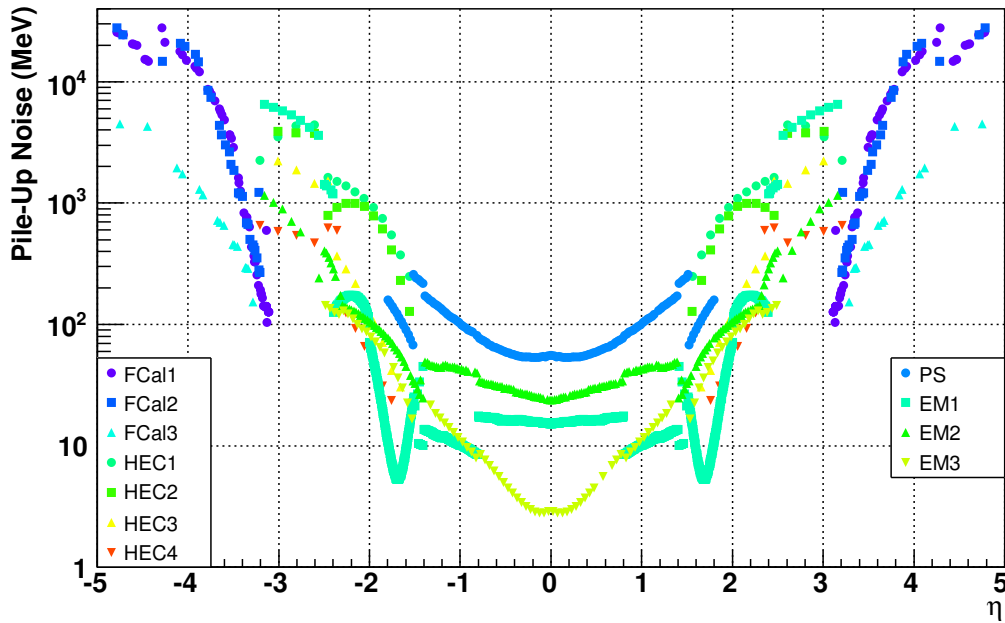


FIGURE 1.7: Bruit d'empilement pour toutes les cellules du calorimètre en gamme haute, à haute luminosité, en fonction de la pseudo-rapacité et pour les différents compartiments en profondeur.

Quelques détails techniques doivent également être précisés :

- Ces calculs sont également nécessaires à la simulation du traitement du signal, ces Tools sont donc également utilisés lors de la digitisation
- CaloNoiseTool et les autres Tools mentionnés héritent chacun d'une interface virtuelle, ce qui permet de découpler l'implémentation de son utilisation.
- Les calculs étant assez lourds, ils ne sont fait qu'une fois et le résultat est caché pour les utilisations suivantes. Les mises à jour sont gouvernées par un mécanisme de call-back. Ce mécanisme permet de spécifier par exemple «Si les Rampes changent (parce que le traitement d'un nouveau run commence) il faut réinitialiser LArADC2MeVTool» ou bien «Si LArADC2MeVTool a été réinitialisé, il faut aussi réinitialiser CaloNoiseTool». Spécifier ainsi les dépendances permet de garantir que si certaines données changent, tout ce qui en découle est recalculé de façon cohérente. Cela permet de traiter plusieurs runs dans le même job, voire de traiter dans un même job des événements issus de plusieurs mois ou même années de prises de données (des candidats boson de Higgs par exemple).

1.3.3 Calorimétrie et trajectographie

Atlas n'est en fait composé que de deux détecteurs : un calorimètre interposé entre deux parties d'un même trajectographe. C'est un raisonnement à la limite en réaction à la situation historique et administrative : les experts du détecteur interne écrivent le code de reconstruction du détecteur interne, les experts muons écrivent le code de reconstruction du spectrographe à muon, et de même pour le calorimètre à argon liquide et le calorimètre à tuile, et à la limite ne se parleraient pas (ces quatre détecteurs ont été conçus et réalisés par des laboratoires différents, avec des structures de direction séparées). Alors que si cette distinction a un sens pour la simulation et pour le décodage des données brutes, elle en a beaucoup moins dès les premières étapes de la reconstruction, car il y a alors autant de différence sinon plus entre les technologies mises en jeu dans un même calorimètre ou trajectographe, qu'entre ces calorimètres ou trajectographes.

1.3.3.1 Trajectographie

Le détecteur interne utilise des pixels de Silicium (dont les clusters fournissent directement des points de mesure tridimensionnels), des bandes de Silicium (qui doivent être combinées pour donner des points de mesure tridimensionnels et ce faisant peuvent donner lieu à des points fantômes), et des tubes à dérives (TRT). Le spectromètre à muon utilise également des tubes à dérive (MDT), mais également des chambres résistives (RPC), des Cathod Strip Chamber (CSC) et des chambres de déclenchement au gaz (TGC). L'orientation du champ magnétique est différente, mais étant donné que ce champ est non uniforme aussi bien dans les toroïdes que (mais dans une moindre mesure) dans le solénoïde, la poursuite des traces dans le champ magnétique doit de toute façon être faite en toute généralité. De nombreux outils et classes de base peuvent donc être partagés. Seule la reconnaissance initiale des traces va être très différente dans les deux cas.

1.3.3.2 Calorimétrie

Le calorimètre argon liquide utilise trois géométries et absorbeurs différents, le calorimètre électro-magnétique (baril et bouchons), le calorimètre hadronique bouchon (HEC) et le calorimètre avant (FCAL). Le calorimètre à tuile fournit la calorimétrie hadronique dans le baril. Tous ces calorimètres ont des cellules projectives en η et ϕ , sauf le calorimètre avant qui a une segmentation en $x-y$ dans le plan transverse au faisceau. Une fois l'énergie de chaque cellule connue (par convention, tous les calorimètres donnent l'énergie à l'échelle électro-magnétique), la reconstruction des clusters, jets ou énergie transverse est similaire quelque soit le calorimètre, si bien qu'une partie significative du code de reconstruction est commune à tous les calorimètres. L'étalonnage précis pourra être différent.

1.3.4 Classe cinématique de base

L'Event Data Model est structuré autour de quelques classes de base, voir Figure 1.8.

- `I4Momentum` décrite ci-après
- `INavigable` permet d'implémenter un système de navigation. Grâce à ce système il est possible de construire des particules composites (par un exemple des jets), et de permettre d'accéder à la liste de ses constituants qui peuvent être de type différents pourvus que ces types dérivent de `INavigable`

- INavigable4Momentum dérive de INavigable et de I4Momentum
- IParticle est une classe de base pour les particules identifiées (apparaissant dans le domaine d'analyse de la Figure 1.5), elle dérive elle-même de INavigable4Momentum

Nous décrivons ici en détail I4Momentum, qui a été développée au LAL, pour en motiver les différents aspects, techniques ou autres.

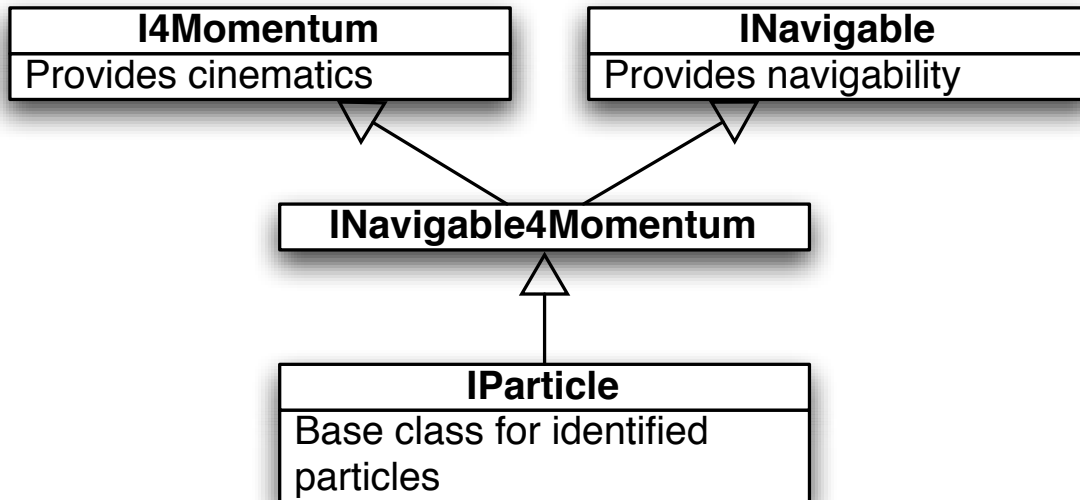


FIGURE 1.8: Classes de base structurant l'Event Data Model, en particulier des AOD

1.3.4.1 Contexte

Les opérations cinématiques, c-à-d sur les quadrivecteurs correspondants aux objets reconstruits sont extrêmement courantes. Il est donc logique de faciliter au maximum ces calculs par un mécanisme commun. Nous détaillons ce mécanisme ici pour expliciter l'intérêt d'une classe de base. Les objections habituelles sur l'utilisation de mécanisme commun sont :

- «Ces calculs sont triviaux»
- «Si on ne sait pas calculer cela, on ne sait pas ce qu'on fait»
- «Je ne fais pas confiance aux outils communs, je préfère faire les calculs moi-même»

Les exemples suivants (cas réels et relativement fréquents même chez des physiciens confirmés) montrent que ces objections ne tiennent pas par rapport à la sécurité donnée par un mécanisme commun :

- Calcul de P_x une composante de l'impulsion d'une trace en fonction du p_T inverse signé de la trace (un paramètre d'ajustement de la trace) et de la direction de la trace à l'origine :

$$P_x = \cos(\phi) / p_{TInv}$$

⇒ Faux ! Il manque la valeur absolue, puisque le p_T est signé par la charge.

- Calcul de l'énergie transverse en fonction de l'énergie et la pseudo-rapidité η :

$$E_t = E \cdot \sin(2 \cdot \text{atan}(\exp(-\eta)))$$

⇒ Correct mais inefficace !, par rapport à $E_t = E / \cosh(\eta)$ qui fait appel à une seule fonction spéciale au lieu de trois (soit environ 300 ns au lieu de 100 ns sur un processeur intel 2.3 GHz cotant 1kSi2K).

- Calcul de l'écart pseudo-angulaire entre deux particules $\Delta R = \sqrt{(\Delta^2\eta - \Delta^2\phi)}$ une mesure de distance couramment utilisée sur les collisionneurs hadroniques :

`DeltaR = sqrt(pow(eta1-eta2,2))-pow(phi1-phi2,2))`

- ⇒ Faux et inefficace ! Si $\phi_1 = \pi - \epsilon$ et $\phi_2 = -\pi + \epsilon$, le ΔR calculé sera $\geq 2\pi$ alors que les deux particules peuvent être en réalité très proches. De plus l'appel à `pow(x, 2)` est beaucoup plus lent que le produit $x * x$: 100 ns contre une fraction de ns.

Les quadrivecteurs généralement utilisés dans Atlas étaient (et sont encore en partie) de la classe `HepLorentzVector` de la librairie CLHEP. Un `HepLorentzVector` est une classe dont les données membres sont `px py pz et e`, et qui est dotée d'un très grand nombre d'accédants cinématiques et de fonctionnalités complexes tels que rotation ou boost de Lorentz.

1.3.4.2 Conception proposée

Le mécanisme proposé [3] est un ensemble de classes de base dessinées Figure 1.9. L'élément clé est la classe `I4Momentum` qui est purement virtuelle : elle se contente de lister un grand nombre de façons d'accéder à un quadrivecteur (les «accédants cinématiques»), avec des noms évidents (`m2()` donne m^2 , `cosTh()` donne $\cos(\theta)$, `hlv()` sera décrit plus loin) sans en implémenter une seule et sans aucune donnée membre. La plupart des objets reconstruits et d'analyse d'Atlas vont dériver indirectement de cette classe. Le premier intérêt est d'uniformiser l'accès aux quantités cinématiques : par exemple que ce soit pour une trace reconstruite ou issue de la simulation rapide, pour un cluster ou un jet la fonction `pt()` retourne l'impulsion transverse, et inversement l'impulsion transverse est retournée par la fonction `pt()`.

Parmi les quatre classes cinématiques de base dérivant de `I4Momentum` actuellement (`P4EEtaPhiMBase`, `P4PxPyPzEBase`, `P4IPtCotThPhiMBase`, `P4PtEtaPhiMBase`), deux ont été représentées parmi les quatre qui existent. Prenons le cas de `P4EEtaPhiMBase` : elle implémente tous les accédants cinématiques de façon optimum en fonction de quatre d'entre eux, dont le choix apparaît très clairement dans le nom de la classe même : `e()`, `eta()`, `phi()` et `m()`, que nous appelons les paramètres «naturels» . `P4EEtaPhiMBase` est abstraite donc sans données membres. `P4EEtaPhiM` est une classe concrète dérivant de `P4EEtaPhiMBase` avec les quatre données membres : `m_e`, `m_eta`, `m_phi` et `m_m` et implémentant les accédants cinématiques correspondants. Par le simple fait de la dérivation, tous les autres accédants cinématiques (`px()`, `cosTh()`, etc...) sont utilisables depuis `P4EEtaPhiM` mais sont en pratique implémentés dans `P4EEtaPhiMBase`.

L'intérêt de cette séparation est de faciliter considérablement pour les objets reconstruits l'implémentation de `I4Momentum`. En effet, suivant les objets reconstruits, les paramètres naturels seront différents mais le nombre de combinaisons est limitée. Dans le cas de `CaloCell` (objet représentant la mesure d'énergie dans une cellule du calorimètre), représentée à titre d'exemple sur la Figure 1.9, les paramètres naturels sont :

- l'énergie mesurée qui varie événement par événement, et qui est donc une donnée membre de `CaloCell`
- η et ϕ donnent naturellement la direction étant donnée la segmentation en $\eta \times \phi$ du calorimètre. η et ϕ ne changent pas événement par événement et sont donc conservés dans un objet statique appelé `CaloDetDescrElement`. `CaloCell` a pour donnée membre un pointeur vers son `CaloDetDescrElement` correspondant, qui permet d'accéder aux données statiques correspondant à la cellule.
- une masse non nulle n'aurait pas de sens

`CaloCell` dérive donc de `P4EEtaPhiMBase` et pour ce faire implémente les quatre accédants

cinématiques naturels. Se faisant, elle s'habille de tous les autres accédants cinématiques, implémentés de façon optimum par rapport à ses paramètres cinématiques naturels. La plupart des objets calorimétriques, pour lesquels e , η , ϕ et m sont des paramètres naturels implémente ainsi tous les accédants cinématiques sans duplication de code. Les objets trajectographie utilisent plutôt P4PxPyPzEBase (historiquement ils utilisaient P4IPtCotThPhiMBase, mais l'ajustement des traces a abandonné le paramétrage des traces au périées en $1/pT$, $\cot(\theta)$, ϕ , a_0 , z qui n'était pas adapté à un champ magnétique fortement non uniforme), les objets liés au déclenchement utilisent plutôt P4PtEtaPhiMBase.

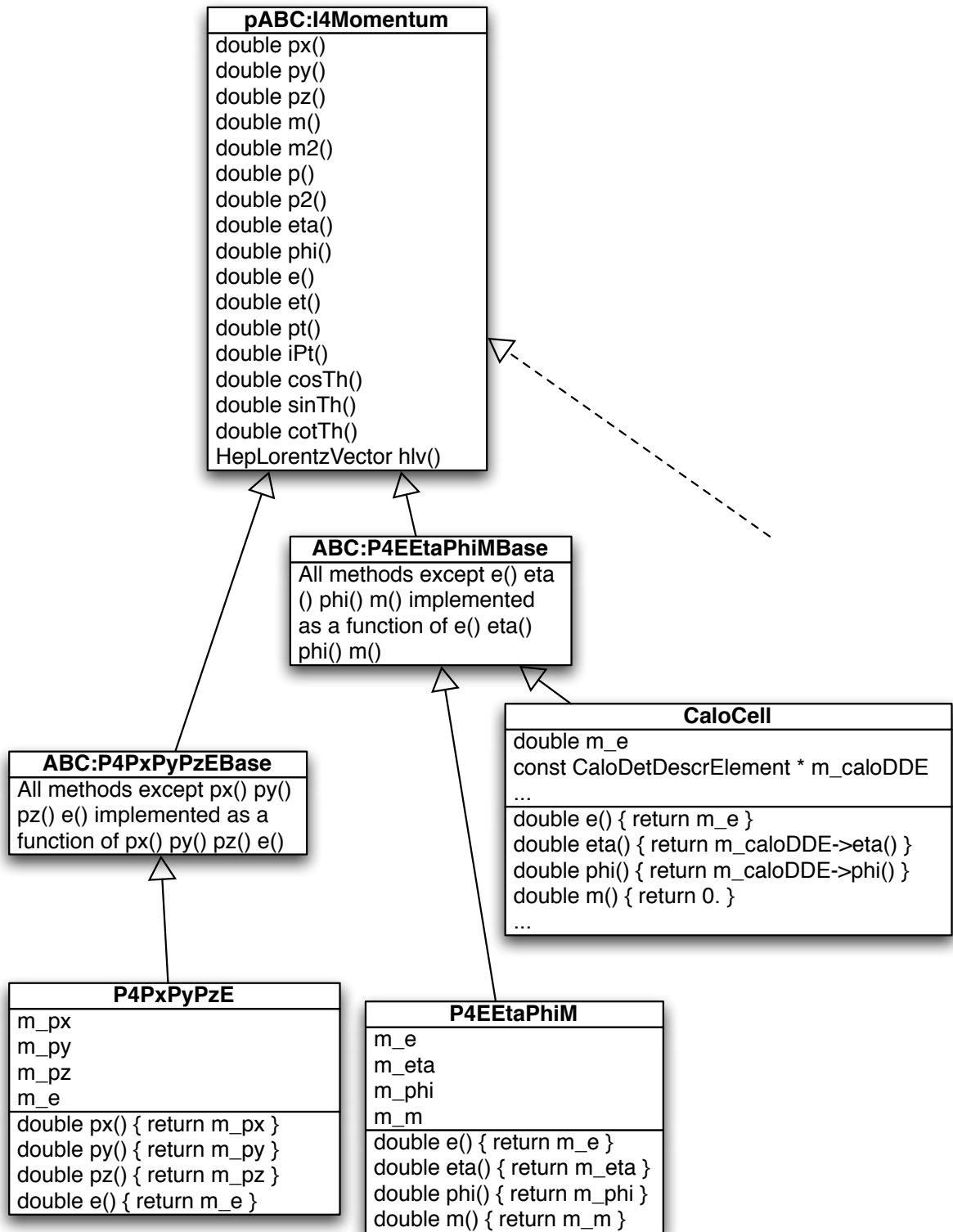


FIGURE 1.9: Classes cinématiques de base. pABC («pure Abstract Base Class») indique une classe de base sans implémentation des fonctions ni donnée membre. ABC («Abstract Base Class») indique une classe de base sans donnée membre. Les flèches indiquent une dérivation de classe, c-à-d. les fonctions et données membres d'une classe de base sont disponibles dans la classe dérivée.

1.3.4.3 Utilisation

Des fonctions d'aide sont également fournies, pour calculer par exemple la masse invariante de plusieurs particules ou leur ΔR . Par exemple, la fonction ΔR est implémentée ainsi :

```
double P4Help::deltaR( const I4Momentum & pA,const I4Momentum & pA)
{ const double deta=pA.eta()-pB.eta() ;
  const double dphi=P4Help::deltaPhi(pA,pB)) ;
return sqrt( deta*deta-dphi*dphi) }
```

où `deltaPhi` est une autre fonction d'aide retournant la différence en ϕ , ramenée à l'intervalle $]-\pi, +\pi]$ comme convention dans tout le code d'Atlas. `deltaR` peut être utilisée comme suit :

```
dR=P4Help::deltaR(theTrackParticle,theJet)
```

où `theTrackParticle` est un objet représentant une trace et `theJet` un objet représentant un jet. Outre la compacité de la formulation, le calcul est fait de manière optimale : le η de `theTrackParticle` est calculé à partir de ses `px()` et `py()` (comme implémentés dans `P4PxPyPzEBase`), alors que l' η de `theJet` est lu directement à partir de sa donnée membre. Inversement, un calcul de masse invariante exigerait de convertir e, η, ϕ et m du jet en e, px, py et pz . Ces conversions se font de façon complètement transparente.

Autre exemple (tiré d'un cas réel) du gain en compacité et en lisibilité, le calcul de la masse invariante de deux candidats électrons (`egamma[0]` et `egamma[1]`), avant l'implémentation de `I4Momentum` :

```
HepLorentzVector hlv[2], hlvZ;
const CaloCluster* clus=0;
double eta,theta,px,py,pz,en;
for ( int i=0 ; i<2 ; ++i ) {
    clus = egamma[i]->get_cluster();
    eta = (clus)->eta();
    theta = 2. * atan(exp(-eta));
    px = (clus)->et() * sin((clus)->phi());
    py = (clus)->et() * cos((clus)->phi());
    pz = (clus)->et() / tan(theta);
    en = (clus)->et() / sin(theta);
    hlv[i].setPx(px); remplissage de chaque quadri-vecteur
    hlv[i].setPy(py);
    hlv[i].setPz(pz);
    hlv[i].setE(en);
}
hlvZ=hlv[0]+hlv[1]; ajout des quadrivecteurs
zMass=hlvZ.mag();
```

après implémentation de `I4Momentum` :

```
zMass=P4Help::deltaM(egamma[0],egamma[1]);
```

ou même si on s'interdit d'utiliser une fonction d'aide :

```
HepLorentzVector hlvZ=egamma[0]->hlv()+egamma[1]->hlv();
zMass=hlvZ.mag()
```

Le gain en compacité et en fiabilité est donc spectaculaire.

1.3.4.4 Compléments d'implémentation

Des points complémentaires doivent être précisés :

- Les classes concrètes (telle P4EEtaPhiM) proposent 4 constructeurs :
 - P4EEtaPhiM() : constructeur vide, les paramètres naturels à zéro
 - P4EEtaPhiM(e,eta,phi,m) : constructeur listant les 4 paramètres naturels dans l'ordre
 - P4EEtaPhiM(const HepLorentzVector & hlv) : constructeur permettant de convertir un HepLorentzVector
 - P4EEtaPhiM(const I4Momentum & p4) : constructeur à partir de n'importe quel objet implémentant I4Momentum
 - P4EEtaPhiM(const I4Momentum * p4) : constructeur à partir d'un pointeur sur n'importe quel objet implémentant I4Momentum

En particulier ces constructeurs autorisent une conversion implicite des paramètres, par exemple P4PxPyPzE theP4(theCaloCell) construit une copie du quadrivecteur de CaloCell sous forme px, py, pz et m .

- Seule les classes concrètes proposent des modificateurs (tels setPx(px) setEta(eta) ...).
 - les modificateurs ne sont pas dans les classes de base, il ne font du sens que pour les classes qui ont des données membres. En effet setEta(eta) n'aurait pas de sens pour une CaloCell étant donné que eta est immuable
 - Seuls des modificateurs des données membres sont proposés, car tout autre modificateur serait ambigu. Par exemple P4EEtaPhiM::setP(p) modifierait-il l'énergie ou bien la masse ? P4EEtaPhiM::setPx(px) effectuerait-il une rotation du vecteur ? Le fait que HepLorentzVector propose de nombreux modificateurs est une source de confusion connue.
- Contrairement à HepLorentzVector, I4Momentum ne propose que des accédants cinématiques. Ce qui est plus complexe ou nécessite de combiner plusieurs quadri-vecteurs est réalisé par des fonctions d'aide, qui sont beaucoup plus facile à étendre sans alourdir l'interface (une autre cas de l'intérêt de la séparation données/algorithmes).
- La connexion avec HepLorentzVector est conservée avec la fonction hlv() qui peut retourner un HepLorentzVector
- Les opérateurs + - n'ont volontairement pas été implémenté dans I4Momentum, car une expression comme theTrack+theJet serait assez ambiguë : il n'est pas du tout évident qu'il ne s'agirait que d'ajouter les quadrivecteurs.
- L'héritage permet beaucoup de souplesse. Par exemple, il a été constaté que le calcul répété de l'énergie transverse pour les quelques 200.000 cellules du calorimètre prenait un temps certain (près de 15% du temps total de reconstruction) uniquement pour les conversions eta vers $\sin(\theta)$. Il a donc été décidé de cacher dans le CaloDetDescrElement le $\sin(\theta)$ de chaque cellule et de réimplémenter l'accédant sinTh() de P4EEtaPhiMBase dans CaloCell, permettant à l'aide de quelques lignes de code dans CaloCell et CaloDetDescrElement d'accélérer significativement, et sans autre modification, les algorithmes d'énergie manquante, de clustering, de jets etc... Cela illustre l'intérêt d'un Event Data Model non trivial, bien que respectant la séparation données/algorithmes.

De même, les méthodes `m2()` et `p2()` ont été ajoutées après coup : en ajoutant quelques lignes dans `I4Momentum`, `P4EEtaPhiMBase` et consœurs ces accédants ont été fournis à toutes les classes implémentant `I4Momentum`. De même, la classe `P4PtEtaPhiMBase` optimale pour le déclenchement a été ajoutée après coup ; d'autres classes sont envisageables.

- L'utilisation d'une classe de base commune `I4Momentum` autorise également la conception de container polymorphique : cela est utilisé en particulier pour la reconstruction des jets, qui doit pouvoir être réalisée en combinant différents objets physiques.

1.3.4.5 Objections

Pour pouvoir être admise dans une grande part du code d'Atlas, il a fallu beaucoup d'itérations pour convaincre toutes les parties prenantes. Certaines discussions n'étaient pas centrées sur des arguments purement techniques :

- Pourquoi ne pas avoir tout simplement hérité de `HepLorentzVector` ?
 - pour ne pas imposer une représentation : utiliser un `HepLorentzVector` (contenant P_x, P_y, P_z, E) pour une cellule du calorimètre pour laquelle E, η et ϕ sont des paramètres naturels auraient exigé des allers et retours perpétuels entre les deux représentations.
 - pour ne pas imposer quatre données membres (par exemple pour `CaloCell` une seule, l'énergie, est une donnée variable d'un événement à l'autre).
 - une méthode de `I4Momentum` et un constructeur dans chaque classe concrète telle `P4EEtaPhiM` permet de passer du système `I4Momentum` au système `HepLorentzVector` qui est toujours utilisé dans `HepMC` et pour des fonctionnalités complexes qu'il serait superflu de vouloir réécrire dans le système `I4Momentum`.
- la masse n'a de sens que pour une particule identifiée, pas pour une trace ou une cellule.
 - En fait, la masse est nécessaire car un calorimètre mesure une énergie tandis qu'un trajectographe mesure une impulsion, passer de l'une à l'autre exige un paramètre de masse. Par convention les traces chargées ont la masse du pion.
- `I4Momentum` étant générique, elle n'autorise pas le changement de référentiel.
 - Ceci est exact, il faut donc des conventions bien établies, qui est que le quadri-vecteur donné est l'estimation la plus précise du quadri-vecteur à l'origine de la particule correspondant à l'objet, basée sur l'objet lui-même : pour une trace, le quadri-vecteur est donné au pégée. pour une cellule, un cluster ou un jet, le quadri-vecteur est donné en supposant que le vertex est à l'origine. Il est possible de fournir des fonctions d'aide (qui ne seraient plus génériques), qui, par exemple, étant donné un cluster et un vertex primaire, fournirait un nouveau quadri-vecteur. D'autre part, il n'est pas question d'utiliser `I4Momentum` si la définition du quadri-vecteur n'est pas intuitive : le quadri-vecteur associé à un cluster dans les pixels n'aurait pas grand sens par exemple.
- l'utilisation d'une classe de base pénalise les performances, car les fonctions d'accès ne peuvent pas être définies comme «inline».
 - Effectivement, chaque appel à un accédant systématique coûte environ 5 ns contre une fraction de ns pour une fonction inline sans classe de base (encore que l'appel successivement à plusieurs accédants du même objet soit moins pénalisant). Mais cette pénalité est négligeable par rapport au gain en simplification.

Toutes ces objections ont pu être levées et I4Momentum avec les autres classes de base décrites au début de cette section est presque systématiquement utilisée dans tout l'Event Data Model de la reconstruction.

1.4 Données persistantes

La définition des données persistantes est à la base de la définition du Computing Model. Quelles sont les différentes étapes du traitement des données, quel volume de données à chaque étape, quel usage pour chaque étape ? La définition des données persistantes est fortement couplée à la définition de l'Event Data Model : il y a une correspondance directe entre les objets transients et les objets écrits, et si en théorie l'écriture des données sur disque ne contraint pas l'Event Data Model transient, ce n'est en pratique pas tout à fait le cas comme nous le verrons.

Il n'y a pas de standard C++ pour l'écriture d'objets sur disque. Il s'agit donc d'une part de définir la technologie employée, d'autre part de définir quelle information écrire et pour quels événements.

1.4.1 Ntuple combiné (CBNT)

Dans le cadre de la préparation au Physics TDR en 1998, il est apparu qu'il fallait rendre plus facile l'accès aux données reconstruites des différents détecteurs, pour pouvoir utiliser Atlas dans son ensemble et non pas détecteur par détecteur comme cela avait été souvent le cas auparavant. Les données persistantes étaient sous forme de banque ZEBRA, mais en pratique ce système était peu utilisé en dehors de la sortie de la simulation et de la digitisation. Dans ce cadre, a été développé un Ntuple Combiné (CBNT en format Hbook [10]) où figurent les informations de la vérité Monte-Carlo, celle des différents détecteurs, les informations combinées (par exemple les paramètres d'un muon sont obtenus en combinant l'information du spectromètre à muon et du détecteur interne) et éventuellement les informations de la simulation rapide. Le code qui remplissait ce ntuple était segmenté en petites routines qui chacune était écrite et maintenue par la personne la plus compétente pour cette partie de la reconstruction. L'utilisateur choisissait par des cartes quelles informations sont nécessaires. Le ntuple était rempli à partir des banques ZEBRA issues des programmes de reconstruction (mais en pratique était le plus souvent créé au vol lors du job de reconstruction). Cela permettait de modifier une analyse rapidement, de pouvoir comparer des algorithmes, de se familiariser facilement avec un nouveau détecteur et d'exporter hors du CERN des données réduites.

Ce même ntuple combiné a été migré dans Athena au fur et à mesure que la reconstruction C++ s'est développé. Cela a permis une comparaison extensive des résultats de la reconstruction fortran et de la reconstruction C++. Athena permettant de basculer du format d'un ntuple Root [11] à Hbook par une simple ligne de commande, cela a permis de soutenir pendant longtemps à la fois les utilisateurs de Hbook et ceux de Root. Le format par défaut est devenu Root à partir de 2004, et le format Hbook définitivement abandonné début 2007 (après les distributions 12.0.6) : en effet l'interface AIDA qui permettait d'écrire aussi bien les ntuple et histogramme Root que Hbook était limitée au plus petit dénominateur commun. Dans la distribution 12.0.6, a été introduit CBNTAA (pour Athena Aware Ntuple) qui écrit directement les Root tree.

Le ntuple ne permet pas de maintenir la richesse des interfaces C++, et la persistance directe des objets C++ a longtemps été étudiée. Le format choisi au départ était une «base de

données orientée objets» Objectivity, qui était utilisée notamment par la collaboration BaBar. Le retour d'expérience de BaBar et les développements expérimentaux au CERN dans CMS comme dans Atlas, n'ayant pas été concluant, un nouveau format séquentiel de persistance des objets «Pool» [12] à été développé par LCG (le groupe de calcul du CERN) à partir de 2003 et adapté à Atlas au cours de 2004. Auparavant, le ntuple combiné (sous format hbook puis root) était le seul mode de persistance de la reconstruction utilisé lors des Data Challenge 0 et 1, c.à.d depuis la migration à Athena.

A partir du moment où on dispose d'un format persistant d'objets (décrit en détail dans la suite), la question se pose si avoir un ntuple combiné qui duplique ces objets avec une interface plus pauvre reste d'actualité. Examinons les inconvénients et les avantages :

- Inconvénient 1 : le fait de disposer d'un ntuple facilement examinable (sous forme texte ou graphique) dans Root, encourage les utilisateurs qui hésitent à passer le pas d'Athena, à développer des macros Root de plus en plus complexes utilisant le ntuple en entrée. Comme l'interface du ntuple est pauvre (presque par définition), et que l'interface C++ complète des objets n'est pas conservée, que les Tools disponibles dans athena ne sont pas utilisables dans Root, chaque utilisateur tend à développer ses propres utilitaires, qui pour changer de représentations, qui pour calculer des masses invariantes etc... Ces utilitaires sont difficilement partageables et, si de nouveaux utilitaires sont écrits sous forme de macro Root, ils devraient être entièrement réécrits pour être utilisables dans Athena.
- Inconvénient 2 : les algorithmes convertissant un objet C++ dans root doivent être maintenus. En pratique, chaque objet est converti par un algorithme séparé, maintenu par une personne compétente. Le travail est donc réparti à de nombreux développeurs.
- Avantage 1 : pour le débogage, un ntuple est immédiatement examinable. Par exemple, la commande

```
CollectionTree->Draw("Trk_pt:PtGen[Trk_truthNt] " , "abs(Type[Trk_truthNt])==211")
```

trace immédiatement la corrélation entre l'impulsion transverse vraie et l'impulsion transverse reconstruite des pions (voir Figure 1.10). La commande

```
CollectionTree->Scan("cl_e_35:cl_e_37:cl_e_35gam:cl_e_37gam")
*****
*   Row   * Instance *   cl_e_35 *   cl_e_37 * cl_e_35ga * cl_e_37ga *
*****
*       0 *       0 * 3113.0769 * 3171.198 * 3007.8991 * 3133.9755 *
*       0 *       1 * 3567.2473 * 3507.3737 * 3397.1430 * 3382.9121 *
*       0 *       2 * 81274.851 * 81476.539 * 81615.039 * 81728.929 *
*       0 *       3 * 15812.152 * 16176.170 * 14891.069 * 14842.191 *
(...)

```

imprime pour les premiers événements l'énergie reconstruite dans le calorimètre électromagnétique dans des clusters de différentes tailles et avec les deux calibrations électron et photon. Une macro Root pour faire des graphes de performance (résolution/efficacité) peut être développé rapidement. Ainsi, les vérifications automatiques de performance dans le système RTT décrites au début de ce chapitre sont le plus souvent basées sur des macros Root utilisant le ntuple combiné. C'est également extrêmement pratique pour investiguer un problème rapporté par un utilisateur, comparer des versions de code différentes, comparé à l'alternative d'écrire un algorithme même simple pour imprimer ou remplir un histogramme. Des développements ont eu lieu récemment pour pouvoir

soit utiliser Root dans athena interactivement, ou bien athena dans Root, mais ils n'ont pas encore complètement abouti.

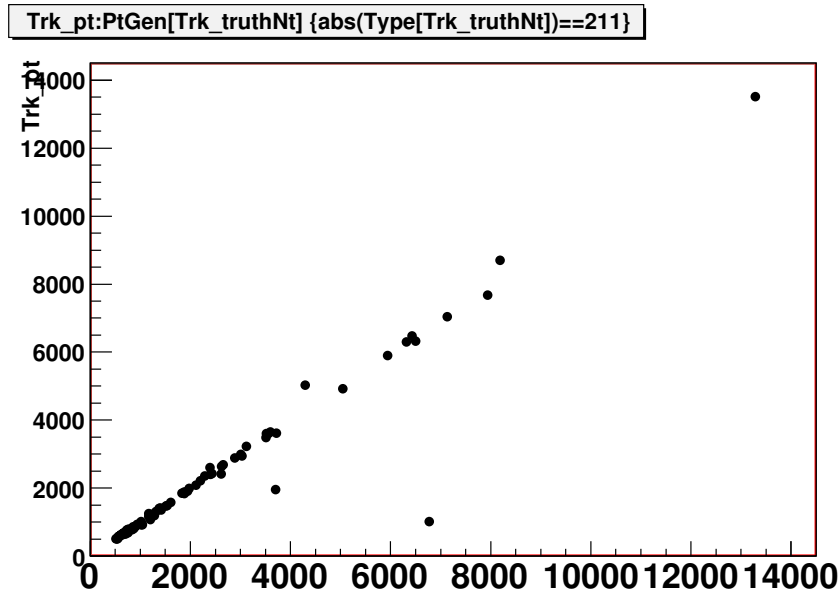


FIGURE 1.10: Exemple de graphe pouvant être réalisé en une seule ligne à partir du ntuplet combiné : impulsion transverse reconstruite des pions en fonction de l'impulsion transverse vraie.

L'avantage 1 est majeur et suffisant pour suggérer de maintenir le ntuplet combiné, du moins tant que la technologie permettant d'accéder directement aux objets écrits n'est pas développée. Dès qu'il sera aussi facile de dessiner un plot ou d'imprimer des variables directement depuis les ESD ou AOD qu'indiqué ci-dessus, le ntuplet combiné n'aura sans doute plus lieu d'être. L'inconvénient 1 ne peut être évité qu'en mettant beaucoup de soin à concevoir les ESD et les AOD, leur documentation, ainsi que les outils associés. Les développements récents liés à EventView et AthenaAware Ntuple, permettent à l'utilisateur de définir ses propres variables (par exemple la masse d'un candidat Higgs) et d'écrire ces variables dans un ntuplet qui pourra être écrit en même temps que les variables CBNT. Ces ntuplets pourront être relus dans Athena. Si bien qu'on peut imaginer un modèle d'analyse où le CBNT est utilisé comme outil de débogage et de vérification de performances.

1.4.2 Vers des objets persistants

La définition des données persistantes est une étape importante car d'une part il s'agit du format qui sera utilisé de façon pérenne, d'autre part il permet de dimensionner les ressources nécessaires à l'exploitation. Une «Task Force AOD/ESD» [13] a été chargée en 2004 de définir les Event Summary Data (ESD) et les Analysis Object Data (AOD), c'est-à-dire la persistance des objets reconstruits dans POOL. Ce qui suit est largement extrait de ce document après mise à jour. Les ESD contiennent la sortie détaillée des algorithmes de reconstruction et contiennent suffisamment d'information pour pouvoir refaire une partie de la reconstruction sans avoir à retourner aux données brutes. Les AOD ont été définies à partir de zéro pour fournir un Event Data Model cohérent pour répondre au besoin de la plus grande partie de l'analyse. Un troisième type, les TAG, permet de faire une sélection rapide des événements intéressants. Des contraintes en tailles doivent être respectées, 500kOctet, 100 kOctet et 1kOctet par événement

respectivement.

Une première version cohérente des AOD et des ESD a été fournie début Septembre 2004 et a été utilisée dans le deuxième Data Challenge. C'est devenu le format standard Atlas, utilisé par la majeure partie de la collaboration, comme l'a montré un sondage réalisé après l'atelier de physique de Rome en Juin 2005. Ce travail a également été utilisé pour définir le Computing Model d'Atlas, c'est-à-dire le flux de données depuis le détecteur jusqu'aux physiciens analystes, chiffrant les besoins en disques, bandes et CPU à tous les niveaux.

1.4.3 *Éléments du computing model*

Le computing model d'Atlas a été établi pour le computing Technical Proposal puis mis à jour pour le computing Technical Design Report [14]. Les grands principes en sont brièvement donnés ici.

Après déclenchement et filtrage en ligne, les données d'Atlas sont écrites à une fréquence d'environ 100 Hz. Sachant que la taille de chaque événement brut est de l'ordre de 1MOctet, cela se traduit par environ 1 POctet (10^{15} octets) par année. Les données brutes (RAW) sont écrites sous forme d'un flux d'octet formatées suivant un format dédié. La reconstruction analyse ces données et écrit les ESD. Une deuxième étape aboutit aux AOD avec les objets prêt à être analysés. Une troisième étape écrit les TAG qui permettront un accès rapide à tout événement. Figure 1.11 indique les différents types de données susceptible d'être écrits à partir de la sortie d'Atlas ou de la simulation. Toutes les étapes, ou transformations, indiquées peuvent être combinées à loisir en un seul processus. Les TAG peuvent intervenir comme moyen de sélection dans n'importe quelle transformation élémentaire, pourvu que l'entrée soit au format Pool, ce qui exclu uniquement la lecture de ByteStream.

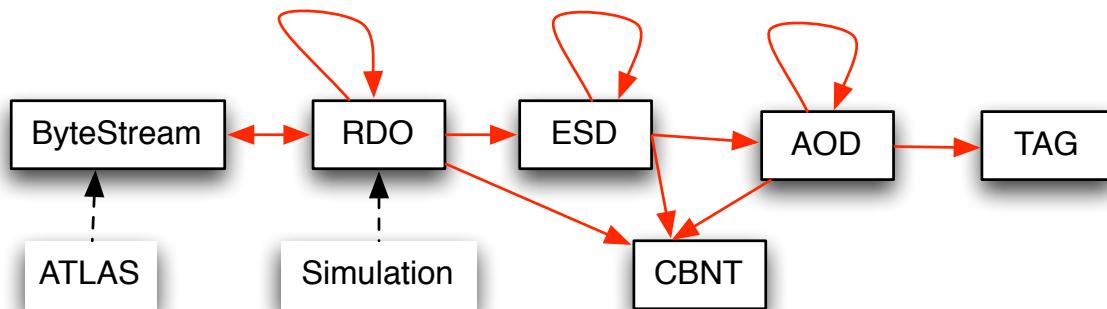


FIGURE 1.11: Différents type de données susceptibles d'être écrites, ainsi que les transformations élémentaires. Les boucles indiquent les copies ou filtres.

Les événements sous forme de flux de bits ou «byte-stream» venant du détecteur (ou bien de la simulation lors de tests préliminaires) sont distribués à une ferme de processeurs pour la reconstruction. La première reconstruction est faite dans le centre calcul dédié du CERN (dit «Tier0»), alors que les reprocessing pourront être distribués. Chaque job de reconstruction écrit les ESD dans son propre fichier d'output POOL. Un run sera réparti sur un ensemble de fichiers.

Les jobs lisant les ESD et écrivant les AOD (aussi en format POOL) sont également exécutés dans le Tier 0. Une dizaine de «streams» AOD peuvent être écrits, pour permettre un

accès rapide aux événements intéressants. La définition de ces streams correspond à des signatures par exemple «au moins un électron de grand P_t », «très grande énergie manquante», etc... Un événement ne figurant que sur un et un seul stream. Il faut noter que ce choix d'exclusivité, est débattu : si il simplifie le book-keeping, il oblige certains streams à être dédiés à la gestion des superpositions des signatures (c.à.d en reprenant l'exemple ci-dessus, gérer les événements avec à la fois au moins un électron de grand P_t et une très grande énergie manquante). Les discussions à ce sujet ont repris en 2006, mais cet aspect n'a aucune influence sur la définition des AOD, et ESD eux mêmes. La définition des AOD est uniforme parmi les streams. Par contre, on peut imaginer des AOD dédiés à une analyse donnée, du point de vue de la sélection comme du contenu.

Lors de la création des AOD, des meta-données (TAG) seront aussi écrites sous format Pool. Le nombre de ces TAG collections, n'est pas encore décidé mais il y en aura au moins une par stream, et certaines pourront combiner plusieurs streams. L'utilisation concomitante de stream et de TAG sera très puissante pour sélectionner les événements, les TAG ayant assez d'information pour pouvoir faire des sélections comme : «deux leptons de p_T supérieur à 20 GeV et une énergie manquante supérieure à 50 GeV», voire même «deux photons de masse invariante entre 100 et 150 GeV». C'est indispensable pour pouvoir présélectionner les événements pour une analyse parmi le milliard d'événements enregistrés par an.

1.4.4 Requis généraux

En général, les AOD doivent donner un accès uniforme aux particules reconstruites (cinématique, variables d'identification...), avec des pointeurs de navigation vers les objets ESD pour une information plus détaillée. La possibilité de pointer directement depuis un objet d'un fichier, vers un objet d'un autre fichier est un atout majeur du format Pool. C'est à dire que, pourvu que le fichier ESD soit présent, l'utilisateur lisant un fichier AOD peut accéder à l'information dans les ESD comme si tous les objets AOD et ESD étaient présents dans le fichier d'entrée.

Les ESD par contre doivent contenir le détail des objets reconstruits, de telle sorte que des raffinements de la reconstruction combinée (identification des particules, réajustement des traces, étalonnage des jets...) puissent se faire depuis les ESD, au fur et à mesure des améliorations des algorithmes ou des étalonnages et alignements. Cela permet un cycle de développement rapide pour la mise au point de ces algorithmes et étalonnages, et permettrait de tester les mises au point sur des échantillons dédiés réduits, avant un reprocessing complet. Néanmoins, on ne saurait espérer refaire depuis zéro le pattern recognition des traces, ou bien recalibrer toutes les cellules, car cela demanderait de sauvegarder trop de données. Les reprocessing complet devront être fait depuis les données RAW.

Les cas d'usage, contraintes et requis suivants ont été identifiés :

- Les objets AOD et ESD doivent pouvoir être disponibles dans le framework d'Athena, éventuellement simultanément.
- Les AOD et ESD doivent pouvoir être écrits avec la technologie POOL
- Un algorithme analysant des AOD, doit pouvoir prendre des ESD en input sans être modifié, et sans recompilation
- Un algorithme analysant la simulation rapide, doit pouvoir analyser les AOD et les ESD sans recompilation
- Il doit être possible d'écrire les AOD et les ESD et les TAG dans le même job
- Les AOD doivent pouvoir être créés depuis les ESD seuls (sans devoir retourner aux

données brutes), ce qui implique :

- Les ESD doivent avoir assez d'information pour cela
- Il ne peut pas y avoir de navigation de ESD vers AOD (seulement de AOD vers ESD)
- Cela peut avoir pour résultat que les AOD créés depuis les ESD et les AOD créés simultanément aux ESD ne sont pas identiques au bit près, en raison de la compactification des objets dans les ESD. Cela est acceptable tant que les différences sont très petites devant la résolution du détecteur. Cependant des effets de seuil sont inévitables, c.à.d par exemple le nombre de particules identifiées dans les AOD peut varier d'une unité.
- Les AOD risquent d'évoluer assez rapidement (en raison d'optimisation des algorithmes d'identification, ajout ou retrait de variables considérées comme indispensables ou inutiles, etc.). Il doit donc être possible de re-générer plusieurs fois les AOD (avec une version différente du software) à partir des mêmes ESD. Cela rend difficile le partage de classes complexes entre AOD et ESD.
- Le budget de taille est 500kOctet pour les ESD et 100kOctet pour les AOD. Toutefois, les premières versions seront probablement plus grandes.
- Toute l'information sur la vérité Monte-Carlo figurent dans des objets séparés. La connexion entre objets reconstruits et vérité est portée par des objets séparés
- La vérité Monte-Carlo contribue significativement aussi bien aux AOD qu'aux ESD, si bien que les événements Monte-Carlo risquent de dépasser largement le budget.
- La navigation des AOD vers les ESD doit être facile, mais contrôlable (pour éviter d'accéder par mégarde aux ESD qui peuvent être sur cassette dans un autre centre de calcul).
- La catégorisation des objets comme «AOD» et «ESD» doit être suffisamment flexible pour qu'elle puisse être révisée dans le futur ou bien pour des productions dédiées.
- Un algorithme navigant vers les ESD doit pouvoir être protégé facilement pour ne pas s'arrêter brutalement si les ESD ne sont pas disponibles.
- Un algorithme qui utilise des informations ESD peut tourner sur les AOD (et naviguer vers les ESD) ou directement sur les ESD de façon transparente et contrôlable
- «reprocessing partiel» : il doit être possible de remplacer un objet d'un AOD ou d'un ESD en le régénérant à partir de l'information existante dans les autres objets, pourvu que l'objet initial n'ait pas été lu. Cela permet d'exécuter les algorithmes d'aval (d'analyse par exemple) sans avoir à les reconfigurer. Toutefois, cela invalidera les objets pointant vers l'objet original. C.à.d en anticipant sur la Figure 1.12 : supposons qu'un utilisateur lise les Photon dans les AOD ; il peut simultanément refaire les objets egamma (suite par exemple à un meilleur réglage des coupures d'identification), et refaire au vol les Photon ; par contre, si les Electron n'ont pas été refaits ils sont invalides.

Le budget en taille a été dépassé délibérément, car il est probable que l'amélioration de la technologie de persistance pourra réduire significativement la taille sans perte d'information. Les AOD/ESD qui ont été proposés satisfont l'ensemble des critères ci-dessus.

La difficulté principale pour la définition des ESD et AOD est d'arriver à un compromis entre la taille sur disque (et dans une moindre mesure à la vitesse de lecture) et à la quantité d'information disponible, qui devra satisfaire le maximum de cas d'usage que ce soit pour les analyses ou pour les travaux d'étalonnage. Pour réduire la taille des ESD et AOD, il y a trois axes d'approche en ce qui concerne les données persistantes :

- réduire la taille à information constante (souvent possible si les objets ne sont pas bien

- conçus au départ)
- réduire la précision des informations jusqu'à un niveau acceptable (la simple précision est suffisante dans la plupart des cas, une compression dédiée peut être conçue dans certains cas (cas des cellules du calorimètre))
- supprimer les informations peu utiles ou redondantes

1.4.5 Contraintes du framework

Cette section décrit les contraintes et fonctionnalités d'Athena qui peuvent avoir une influence sur les objets AOD et ESD. La liste des objets écrits dans les AOD et ESD est donnée comme une liste d'items pour chaque AthenaOutputStream, chaque item correspondant à un objet du Transient Event Store. Pour le même job, différents AthenaOutputStream peuvent être configurés, chacun avec sa propre liste d'items. Pour chaque objet dans la liste, un convertisseur assurant la persistance de l'objet est appelé à la fin du traitement de l'événement. Pour la plupart des objets, ce convertisseur est générique, c.à.d automatiquement généré depuis le header C++ de l'objet, avec de petites modifications potentielles indiquées par le fichier `selection.xml` (qui peut par exemple indiquer qu'une donnée membre ne doit pas être écrite). Il est possible également d'écrire manuellement un convertisseur quand il faut utiliser une compression spéciale. Il y a toutefois deux contraintes importantes pour les convertisseurs :

- il ne peut y avoir qu'un convertisseur pour un type d'objet donné, c.à.d une seule façon de persistifier un type d'objets. Les convertisseurs manuels peuvent être configurés, donnant plus de souplesse, mais pour un job donné ils ne peuvent être configurés que d'une façon donnée. Donc, comme les AOD et ESD doivent pouvoir être écrits dans le même job, pour un objet qui figurerait à la fois dans les AOD et les ESD, il n'est pas possible de définir deux représentations persistantes différentes. Par exemple, pour un objet Track complet, on ne peut pas l'écrire d'une part intégralement dans les ESD, et d'autre part seulement les paramètres cinématiques dans les AOD. C'est par contre possible en divisant un objet conceptuel en plusieurs objets C++, voir Section 1.4.6.
- un convertisseur donné ne traite qu'un objet à la fois. Il n'est pas possible de fondre plusieurs objets en une seule représentation persistante sauf dans un cas particulier : un objet transient peut être reconstruit à partir de la représentation persistante d'un autre objet. C'est par exemple le cas actuellement avec les tours du calorimètres qui sont recalculées au vol à partir des cellules. Il faut toutefois être très prudent avec cette approche : supposons qu'un objet B puisse être déduit d'un objet A et que les deux soient écrits dans les ESD pour la distribution n. Dans la distribution n+1, le Tool qui déduit B de A a été amélioré ou débogué. Supposons qu'on relise un ESD de la distribution n avec la distribution n+1 : l'objet B recréé au vol ne sera pas le même que l'original, et l'ensemble des objets lus ne sera pas cohérent.
- les relations entre objets sont difficiles à persistifier. Des relations génériques comme un pointeur d'un objet à un autre ou à un objet dans un container sont traitées par les `DataLink` ou les `ElementLink`, qui sont des pointeurs persistifiables.

1.4.6 Définition des AOD et ESD

Pour définir les ESD, l'approche a été de partir de l'Event Data Model complet de la sortie des algorithmes de reconstruction, et de supprimer ou compresser les objets les plus gros ou inutiles. Par exemple, les clusters et point d'espace du détecteur interne ont été supprimés (ce qui empêche de refaire la recherche de traces depuis l'ESD), mais pas les `RIO_OnTrack`, qui

sont les clusters associés aux traces, et qui permettent donc de réajuster les traces si besoin. Toutefois la possibilité d'écrire ces informations pour des études particulières d'étalonnage ont été conservés. En ce qui concerne les cellules du calorimètre (CaloCell), un convertisseur spécial a été écrit permettant de les compresser avec une faible perte d'information.

Par contre la définition des AOD a été faite de zéro pour présenter l'information au niveau de l'analyse de la façon la plus homogène possible. Il n'est pas question de donner ici le contenu détaillé des classes persistantes, mais de donner quelques indications générales avec quelques exemples. La Figure 1.12 donne le diagramme simplifié de répartition des objets dans les AOD et les ESD. Les flèches indiquent les pointeurs persistants, qui en général reproduisent le flux de données. Les noms sont les noms des classes C++ qui sont relativement clairs. Lorsque plusieurs instances de la même classe, la clé est précisée (par exemple «Track/ID» et «Track/Muon»). Le diagramme est simplifié car :

- Les objets effectivement écrits sont dans la plupart des cas des collections (ou conteneurs) des objets indiqués sur le diagramme
- les objets annexes formant un tout consistant avec les objets indiqués sont omis
- lorsque des algorithmes alternatifs existent seuls un objet est indiqué (par exemple il y a plusieurs collections de jet pour les algorithmes Kt et cône, et avec différents paramètres)
- les objets concernant la vérité ont été omis

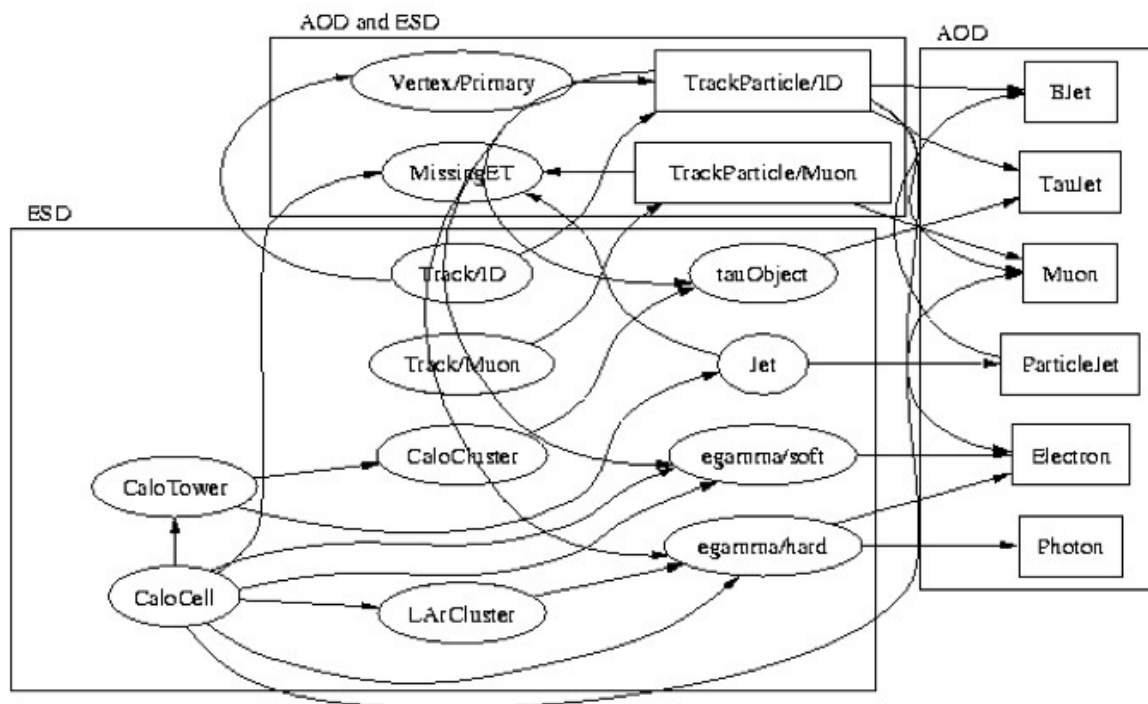


FIGURE 1.12: Diagramme simplifié de répartition des objets dans les AOD et les ESD. Le rectangle intitulé «AOD and ESD» indiquent les objets qui sont copiés des ESD vers les AOD. Les flèches indiquent les pointeurs persistants.

La multiplicité des flèches indique clairement que l'information détaillée sur la composition d'un objet final est préservée dans les AOD, ou bien dans les ESD. En prenant un exemple précis : l'Electron (AOD) pointe sur l'egamma et sur le CaloCluster (ESD) électromagnétique

dont il est issu et sur la `TrackParticle` (ESD et AOD) associé, le `CaloCluster` pointe sur les `CaloCell` (ESD) qui le comportent, la `TrackParticle` pointe sur la `Track` (ESD) qui comprend les impacts dont il est formé. En général pour faire l'analyse, un physicien se contentera d'accéder à l'`Electron`, mais il peut aussi, par exemple pour un événement pathologique, en utilisant les mêmes données examiner le détail de la reconstruction de la trace de l'électron, ou bien même refaire l'identification des électrons. Naturellement, tous les physiciens n'auront pas à un accès direct aux ESD pour tous les événements (puisque l'un des buts des AOD est précisément d'avoir un format compact pour maximiser la statistique disponible), mais il est assez facile pour une analyse donnée de procéder à une sélection des événements intéressants.

L'information sur un même objet conceptuel peut être répartie entre les AOD et les ESD. Par exemple pour une trace, la `TrackParticle` (figurant à la fois dans AOD et les ESD) contient l'information sur les paramètres de la trace au péricée (donnant accès au quadri-vecteur), et des informations sur la qualité de la trace (χ^2 , nombre d'impacts dans les différents détecteur, information sur l'identification dans le TRT). La `TrackParticle` pointe sur la `Track`, qui comporte la liste de tous les impacts associés à la trace, ainsi que les paramètres de la trace au niveau de chaque impact. Ce schéma a été adopté par la suite pour d'autres objets comme les `CaloCluster` : l'objet `CaloCluster` (dans AOD et ESD) ne contient outre le quadri-vecteur que quelques paramètres tels l'énergie dans chaque couche du calorimètre, mais pointe sur d'autres objets dans les ESD avec la liste des cellules qui le composent et des paramètres de forme détaillés.

1.5 La grille

Le noeud principal sera le Tier 0, un centre de calcul du CERN dédié au traitement immédiat des données dans un délai inférieur à 24 heures. Les reprocessings seront assurés par une dizaine de Tier 1, les centres de calcul régionaux dont fait partie le centre de calcul IN2P3-CEA/DAPNIA de Lyon. Les ESD et les données brutes correspondants à un run donné sont envoyés sur deux Tier 1 qui en assurent la garde. Ainsi, chaque Tier 1 garde un échantillon non-biaisé d'un cinquième des données. Par contre, l'intégralité des AOD et des TAG sont distribués à tous les Tier 1. Il existe une cinquantaine de Tier 2, centres de calcul locaux, assurant d'une part les jobs d'analyse et d'autre part les productions Monte-Carlo. Ces réseaux de centre forment la grille de calcul («the Grid»), même si en pratique il existe trois grilles indépendantes mais avec des efforts d'inter-opérabilités. La grille est plus qu'un super-système de batch.

Le premier défi qui a été résolu depuis 2005 environ est de pouvoir distribuer des jobs sur des systèmes très divers distribués dans le monde entier. Pour cela, un système de définition des jobs universel a été adopté, mais la difficulté principale fut la distribution des logiciels Atlas. Pour chaque distribution (même les distributions de nuit), un «kit» est construit qui contient tous les fichiers nécessaires à l'exécution de n'importe quel job : binaires, scripts de configurations, fichiers et base de données de étalonnages. Ces fichiers ne sont pas tous développés par Atlas, même les logiciels développés à l'extérieur (e.g. Geant4, logiciels LCG dont Pool, et même le compilateur C++) figurent dans ce kit : c'est le seul moyen pour être certain que la version correcte de ces logiciels soient installés sur tous les centres de calcul. L'exportation du kit dans les centres de calcul est faite de manière automatique : néanmoins une surveillance active de ce processus est toujours nécessaire. Le kit contient également les headers C++ ce qui permet de développer du code ou de l'analyse.

Le deuxième défi est celui de la puissance de calcul : il était en passe d'être relevé début 2007, car à ce moment là, environ 50.000 CPU étaient potentiellement disponibles pour les jobs

Atlas sur la grille (principalement pour la simulation), alors que 10.000 suffiraient à absorber la reconstruction immédiate des événements d'Atlas (ceci écrit pour donner un ordre de grandeur, sachant que la reconstruction immédiate des événements sera entièrement réalisée au Tier 0 et non pas sur la grille). L'inconnue subsistant concerne les besoins pour les jobs d'analyse : en général le physicien a tendance à utiliser pour son analyse toute la capacité disponible et à réfléchir ensuite : un système de quota devra donc être imposé.

Le troisième défi à relever est celui de la gestion des données. En effet, pendant la simulation à grande échelle encore en cours début 2007, il n'est pas rare de passer chaque jour 10.000 jobs de simulation de 50 événements chacun, soit un volume de données de près de 1 TOctet par jour ! Ces données doivent être rapatriées de telles sortes que des ensembles de données complets (correspondant chacun à un canal Monte-Carlo particulier) soient disponibles dans les Tier 1 et 2 avant la reconstruction. C'est la gestion de ces données qui n'est pas encore fiable et qui devra être significativement améliorée au cours de l'année 2007.

2 ALGORITHMES DE RECONSTRUCTION POUR ATLAS

Après une présentation rapide d'Atlas cette section fait un tour d'horizon des algorithmes de reconstruction existants (avec pour point de référence la release 12.0.6, disponible début 2007). Il n'est pas question de donner ici une revue exhaustive des performances des différents algorithmes mais de mettre l'accent sur les développements les plus récents et futurs, et en décrivant plus en détail si nécessaire les détecteurs. Les quelques chiffres cités sont extraits de [15]. L'exercice CSC en cours début 2007, avec simulation et reconstruction à grande échelle, permettra une mise à jour détaillée des performances des algorithmes, et permettra une dernière réoptimisation avant le début de la prise des données.

2.1 Reconstruire ou la résolution d'un problème inverse

Lors d'une collision proton-proton, un ensemble de particules se propagent à travers le détecteur en laissant un signal dans les différentes parties actives et en interagissant avec la matière (diffusion multiple pour les traces chargées, rayonnement de freinage pour les électrons, conversion pour les électrons, puis gerbe électromagnétique ou hadronique dans les calorimètres, etc...). Ces signaux sont des signaux électroniques qui sont mis en forme par l'électronique frontale et finalement écrits sous forme digitale, les données brutes.

La reconstruction, ou en bon français la reconstitution, est l'ensemble des algorithmes permettant de résoudre le problème inverse : remonter à partir des données brutes à la meilleure connaissance possible sur les particules initiales :

- identité des particules : photons, électrons, muons, tau, hadrons (éventuellement identification pions/kaons/protons mais Atlas ne dispose pas des détecteurs adéquats pour cela), jets et b-jets (qui ne sont pas à proprement parler des particules, mais un ensemble de particules qui n'ont pu être nécessairement identifiées séparément), neutrinos (indirectement par l'énergie transverse manquante)
- estimation de leurs quadrivecteurs impulsion et si possible de leur vertex d'origine
- comme l'identification et la mesure ne sont jamais parfaites, et comme les besoins pour les différentes analyses ne sont pas les mêmes, des estimateurs de la qualité de l'identification et de la reconstruction sont nécessaires.

Les Figures 3.31 page 69 et 3.37 page 80 donnent des représentations d'événements simulés où figurent photons, électrons, muons et jets. Les points et rectangles représentent les signaux bruts après pré-reconstruction (voire ci-dessous), les lignes courbes les trajectoires reconstruites (la reconstruction des jets n'est pas indiquée).

Il s'agit donc d'extraire des quelques 10^8 canaux électroniques lus, les paramètres de la centaine de particules identifiables typiquement présentes dans un événement, qui soit utilisables pour l'analyse. Chaque particule peut laisser un signal dans un élément du détecteur et il n'y a pas d'information a priori sur quels éléments du détecteur ont été touchés par une particule donnée. Le problème pourrait donc donner lieu à une combinatoire quasi insoluble si la diversité de ce qui est recherché n'était pas relativement faible : il s'agit de reconstruire les particules issues du point d'interaction (lui-même contenu dans une zone lumineuse aux dimensions limitées), et pour lesquelles est connu pour chaque type un modèle de la façon de se propager et d'interagir dans le détecteur. L'inconvénient, mesuré, en est qu'un objet produit se propageant de façon irrégulière ne serait pas reconstruit, donc détecté.

Un autre aspect qui rend le problème solvable est la factorisation : un électron aura le même type de trajectoire pour tout événement, un point de mesure dans les pixels sera similaire

quelle que soit la nature de la trace chargée, même si la densité de particules au voisinage peut altérer en partie la reconstruction d'une particule donnée. La reconstruction procède en général par étapes, qui sont dans les grandes lignes :

1. pré-reconstruction : il s'agit d'un premier passage permettant de traduire le signal électronique d'un élément du détecteur (ou de quelques éléments voisins) en un point de mesure utilisable : pour les trajectographes, point de traversée de la particule avec l'incertitude estimée, pour les calorimètres énergie dans une cellule rapporté à l'échelle électromagnétique. Cette pré-reconstruction peut avoir été en partie réalisée en-ligne.
2. pattern recognition ou en français reconnaissance de formes : probablement la partie la plus difficile, il s'agit d'associer entre eux, pour un détecteur donné, tous les signaux provenant d'une même particule : les points de la trajectoire ou les cellules de la gerbe calorimétrique. Les qualités essentielles du détecteur facilitant le pattern recognition sont :
 - la granularité : plus elle est fine, moins il est probable qu'un élément du détecteur aura enregistré le passage de deux ou plus particules superposées ;
 - la précision : plus elle est grande plus faible sera la combinatoire pour associer les points de mesure en des trajectoires valides.
3. ajustement des paramètres de la particule étant donné les éléments de mesure. Il peut être fait simultanément à un pattern recognition dit local, où l'association de certains points de mesure à une particule peut être revus.
4. identification des particules, en général en combinant les informations de plusieurs détecteurs (par exemple combinaison calorimètre électro-magnétique/trajectographe interne pour identifier les électrons)
5. ajustement final des paramètres de la particule, en général en combinant les informations de plusieurs détecteurs et sous l'hypothèse de l'identité la plus probable (par exemple les étalonnages à appliquer aux électrons et aux photons sont différents).

2.2 Atlas dans une coquille de noix

Le détecteur Atlas (pour une description détaillée voir par exemple [1]), est un détecteur polyvalent optimisé pour analyser les collisions proton proton à haute énergie (14 TeV dans le centre de masse) au LHC. Il a une géométrie classique de cylindre fermé par des bouchons, un diamètre de 22 m et une longueur de 44 m (voir Figure 2.13). Sa densité moyenne est inférieure à un, si bien qu'Atlas, telle une coquille de noix, flotterait...

Le détecteur interne («Inner Detector») baigne dans un champ magnétique de 2 T (légèrement non uniforme aux extrémités), son acceptance s'étend jusqu'à $|\eta| = 2.5$. Au-delà du solénoïde le calorimètre à Argon liquide fournit une calorimétrie électro-magnétique dans le baril et une calorimétrie à la fois électro-magnétique («LArEm») et hadronique («HEC») dans les bouchons. Le calorimètre électromagnétique s'étend jusqu'à $|\eta| = 2.5$ et la calorimétrie totale s'étend grâce aux calorimètres avant (FCAL) jusqu'à $|\eta| = 5$. La calorimétrie hadronique dans le baril est donnée par un calorimètre fer-tuiles scintillantes («Tile»). Ces éléments sont entourés par les toroïdes à air du spectromètre à muon : un toroïde baril et deux toroïdes bouchons.

2.3 Empilement

En raison de la section efficace totale p-p élevée, et malgré le nombre élevé de paquets dans le LHC, à chaque événement de physique se superpose en moyenne 2,3 événements de biais minimum à basse luminosité $L = 10^{33} \text{cm}^{-2} \text{s}^{-1}$, et 23 à haute luminosité $L = 10^{34} \text{cm}^{-2} \text{s}^{-1}$.

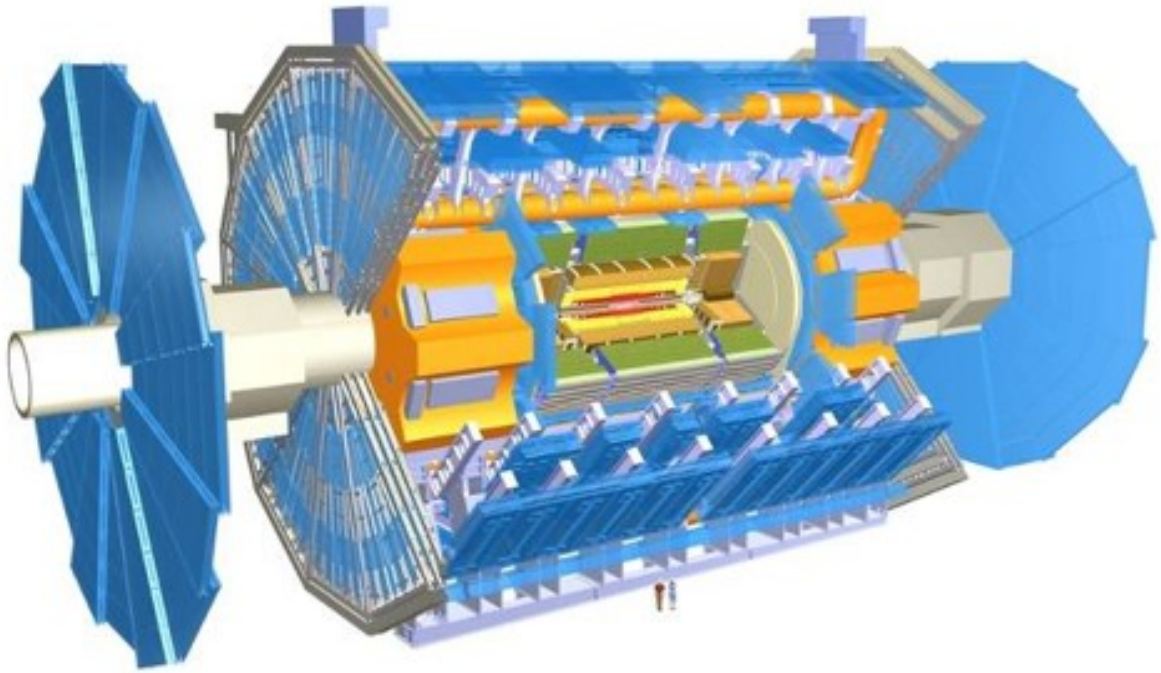


FIGURE 2.13: Le détecteur Atlas

Ce fait, ainsi que le faible laps de temps (25 ns) entre deux collisions a été une contrainte majeure sur la conception du détecteur et a également un impact sur les algorithmes de reconstruction.

Les événements de biais minimum ont une faible multiplicité de particules (voir Figure 2.14), de faible énergie transverse. Au niveau du générateur un nombre de particules chargées d'environ 7 par unité de pseudo-rapidité, donne 35 particules dans l'acceptance du tracker (environ le double dans l'acceptance totale), soit 800 pour 23 collisions. Le nombre de particules neutres est un peu supérieur, et comme il s'agit principalement de photons, leur conversion possible dans la partie interne du détecteur accroît encore jusqu'au delà de 1000 le nombre de particules chargées traversant le détecteur. Ces particules chargées sont de faible impulsion transverse et la plupart d'entre elles ne peuvent pas être reconstruites. Cependant, elles sont sources de nombreux impacts superflus dans les différents éléments du tracking. De plus, le temps de dérive dans les pailles du TRT de 40 ns est tel que le croisement de faisceau immédiatement antérieur ou postérieur au croisement intéressant peuvent contribuer significativement. Ce n'est pas le cas des Pixels et SCT car le temps d'intégration dans le Silicium est très rapide.

En ce qui concerne le calorimètre à Argon liquide, le signal durant près de 400 ns a été accéléré pour ne couvrir que quelques croisements de faisceau. Figure 2.15 montre la mise en forme bipolaire du signal. L'intégrale en est nulle si bien que le signal des événements de biais minimum de toutes les collisions est en moyenne nul, même si le bruit résultant est significatif. Les 5 points de part et d'autre du pic sont mesurés, permettant une meilleure évaluation de

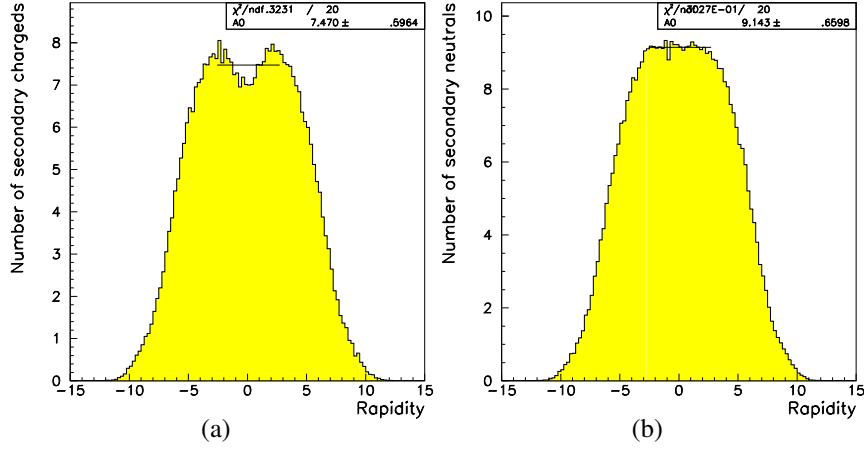


FIGURE 2.14: Nombre moyen de particules chargées (à gauche) et neutres (à droite) par unité de pseudorapacité, dans les événements de biais minimum simulés avec Pythia 5.7 et utilisés pour le Physics TDR [1]

l'énergie déposée à l'aide d'une méthode de filtrage optimal. En ce qui concerne le calorimètre à tuile, la forme est similaire mais toujours positive, et 7 points sont mesurés.

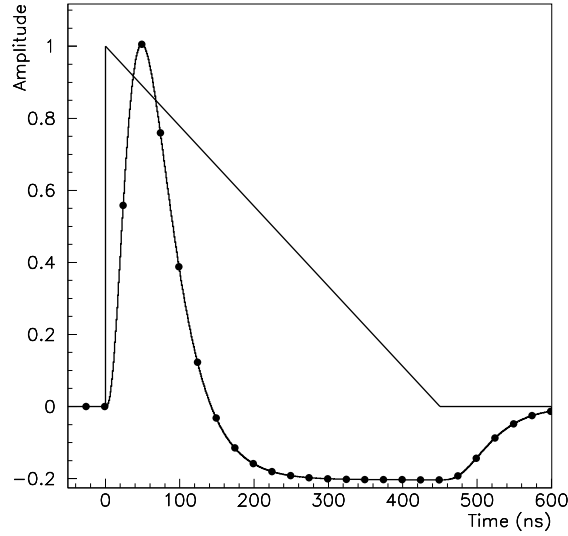


FIGURE 2.15: *Forme du signal, brut (triangulaire) et après mise en forme dans les FEB (courbe bipolaire); les points représentent l'échantillonnage toutes les 25ns*

Les détecteurs à muons ne souffrent pas directement de l'empilement car ils sont protégés par les calorimètres. Toutefois, les interactions multiples dans le détecteur donnent en bout de chaîne à une sorte de gaz de particules de très faibles énergies (quelques MeV) qui remplit toute la caverne et donne lieu dans les chambres à muons à des taux de comptages significatifs, mais non corrélés avec le croisement de faisceau,

2.4 Tracking interne

La reconstruction des traces dans le détecteur interne est réalisée jusqu'à une pseudo-rapidité de 2.5 avec trois couches de pixels (dont la plus interne, la b-couche, contribue le plus à la précision du paramètre d'impac), puis quatre couches de détecteur silicium à bandes stéréo, suivi par un détecteur à pailles donnant typiquement 30 mesure par temps de dérives, en plus d'une identification des électrons par rayonnement de transition des électrons. La résolution en paramètre d'impact transverse approche 10 microns à grand P_T , mais est dégradée significativement pour les P_T de 1-10 GeV par la diffusion multiple.

La résolution en impulsion est dominée par la diffusion multiple jusqu'à 20 GeV donnant une résolution de 1.5%. La présence d'empilement à haute luminosité ne dégrade pas significativement les performances de la trajectographie, car le trajectographe a été optimisé pour séparer les traces dans les jets de grand P_T où la densité de particules reste bien plus grande que celle de l'empilement.

2.4.1 Développements récents et en cours

Le code C++ a été entièrement réécrit sous forme de modules pour la reconnaissance des traces, l'ajustement, l'extrapolation, la résolution des ambiguïtés. La plupart de ces modules existent sous plusieurs versions. La principale piste de développement actuelle est de choisir dans cette boîte à outils la combinaison optimale, en terme d'efficacité de reconstruction, précision de reconstruction, absence de fausse trace, robustesse, mais aussi de rapidité et d'utilisation de la mémoire. Ce choix ne sera probablement pas unique : dans les premiers mois, la robustesse par rapport aux erreurs d'alignement sera un paramètre important, par la suite la robustesse par rapport aux interactions multiples le deviendra.

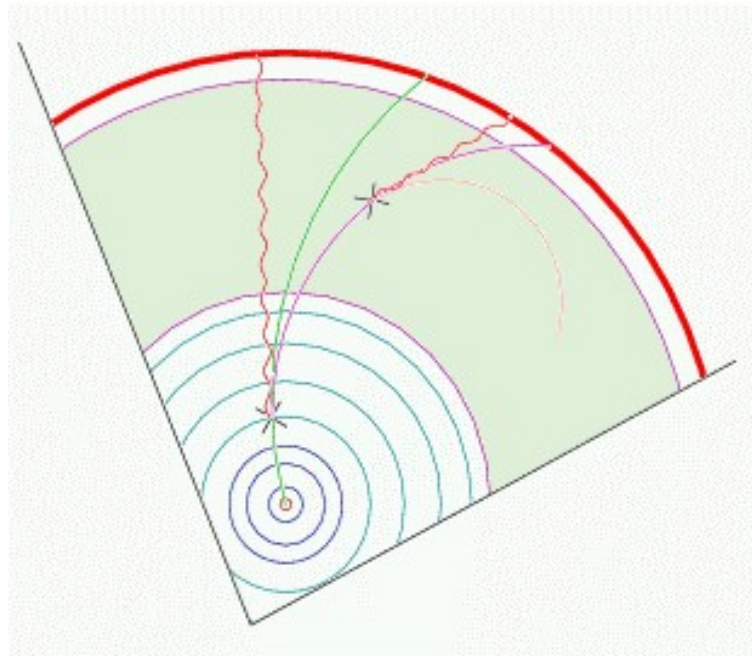


FIGURE 2.16: Illustration de l'émission de photons de Bremsstrahlung par un électron traversant le baril du détecteur interne. La trajectoire des deux photons émis est indiquée par une ligne ondulée. Outre la trajectoire de l'électron sont indiquées également les extrépolations et l'absence d'émission de photon.

Le développement d'une reconstruction de trace dédiée aux électrons est nécessaire en rai-

son de l'importance de la longueur de radiation intégrée dans le trajectographe. La Figure 2.16 illustre la particularité de la trajectoire d'un électron perdant de l'énergie par Bremsstrahlung dans le trajectographe. L'idée est d'utiliser ce tracking dédié uniquement devant les clusters électromagnétiques préalablement identifiés. Plusieurs approches sont nécessaires :

- le rayonnement de freinage mou se traduit par une perte d'énergie régulière. Cette perte d'énergie peut être traitée de la même manière que la perte d'énergie par ionisation pour les autres particules chargées, c.à.d en estimant l'augmentation moyenne de la courbure au passage de chaque élément de matière.
- le rayonnement de freinage dur se traduit par une cassure dans la courbure des électrons qui peut être ajustée en ajoutant deux paramètres à l'ajustement de la trace : le rayon où l'électron est émis et l'énergie du photon émis. Même si en principe plusieurs photons durs peuvent être émis, il n'y a pas assez de points de mesure pour en contraindre plus de un. De même, en ajoutant des paramètres, l'ajustement de la trace est rendu moins discriminant envers les fausses traces, d'où l'intérêt de tenir compte du cluster électromagnétique en tant que contrainte supplémentaire.
- en cas de rayonnement dur, la trace peut ne pas être trouvée du tout par le pattern recognition standard. Un pattern recognition dédié partant du cluster électro-magnétique pourrait être développé, ce serait surtout utile pour les électrons issus de conversion de photon.

2.5 Vertexing

Les algorithmes de vertexing comprennent deux parties relativement distinctes, l'ajustement du vertex et l'assignation des traces au vertex.

2.5.1 Ajustement du vertex

L'ajustement du vertex est un problème non trivial mais relativement bien défini : étant donné un certain nombre de traces, quel est le point (donné par ses coordonnées et leur matrice de covariance) de l'espace le plus compatible comme étant l'origine de ces traces. Les caractéristiques de l'algorithme idéal sont :

- Estimation non biaisée des coordonnées du vertex
- Estimation correcte de la matrice de covariance des coordonnées (Bien sur, cela dépend en grande partie du fait que les traces elles-mêmes aient une matrice de covariance correcte.)
- Fourniture d'un χ^2 se comportant bien, pour permettre de tester la validité du vertex.
- Fourniture de la contribution au χ^2 de chaque trace
- Ajustement du vertex seul (les traces étant fixes), ou bien ajustement du vertex avec réestimation des paramètres des traces. Cette dernière possibilité est plus lente, parfois un peu plus précise en ce qui concerne le vertex, mais surtout peut améliorer la précision sur les paramètres des traces, et permet d'avoir une estimation optimale du quadri-vecteur d'impulsion total.
- Possibilité de la prise en compte d'une contrainte de position. L'exemple typique est de contraindre le vertex primaire à être dans la zone lumineuse.
- Prise en compte correcte de l'erreur due à la diffusion multiple : les paramètres des traces sont donnés soit au point de mesure le plus interne, soit extrapolé au péricée en tenant compte de la diffusion multiple dans le tuyau du faisceau et les matériaux proches. Dans les deux cas, il faut qu'une fois une première estimation du vertex

connue, les traces soient extrapolées convenablement jusqu'à ce vertex, sinon leur erreur risque d'être trop grande ou trop petite, donc leur poids dans le calcul du vertex mal ajusté, et la matrice de covariance fausse. C'est particulièrement important pour les vertex détachés comme les conversions ou les désintégrations de K_s^0 ou Λ .

- Ajustement dédié pour les vertex de conversions : les deux électrons venant d'une conversion sont parallèles au point de conversion, si bien qu'à moins d'une paramétrisation spécifique l'ajustement du vertex peut être instable (dérivée seconde du χ^2 nulle)
- Possibilité d'ajuster des vertex en cascade : l'exemple type est la désintégration $B \rightarrow J/\psi(\mu^+\mu^-)K_s^0(\pi^+\pi^-)$, où le vertex du K_s^0 peut être à plusieurs dizaines de centimètres du vertex du B^0 . L'ajustement peut être simultané ou bien séquentiel : le K_s^0 est ajusté sous forme d'une pseudo-trace qui est dans un deuxième temps combiné aux deux muons.
- Possibilité de prendre en compte des «traces neutres» : il doit être possible d'inclure un photon mesuré dans le calorimètre dans le vertex.
- Possibilité d'émettre des contraintes de masse : dans l'exemple ci-dessous, les deux muons peuvent être contraints tels que leur quadri-vecteur somme ait la masse du J/ψ .
- Possibilité d'ajouter ou supprimer facilement une trace sans reprendre tout l'ajustement, ce qui est possible en utilisant une méthode de filtre de Kalman.

Pendant longtemps, seul un code fortran CTVMFT venant de CDF a été utilisé. De nouveaux Tools C++ sont développés mais chacun n'a pas toutes les fonctionnalités mentionnées ci-dessus.

2.5.2 Assignment des traces aux vertex

Lors de la reconstruction d'un K_s^0 ou d'un canal de désintégration de hadron beau, les traces à inclure dans le vertex sont bien définies (même si l'ajustement du vertex fait partie de la procédure pour de validation de l'assignation des traces aux vertex). Par contre, il est des cas où on ne sache pas quelles traces appartiennent à quel vertex. Les exemples typiques sont la détermination du vertex primaire et la détermination inclusives du vertex secondaire du hadron B (voire du vertex tertiaire du hadron charmé) dans le cadre de l'étiquetage des jets de b , qui est détaillé Section 2.10.

En ce qui concerne le vertex primaire, un aspect important est que celui-ci est inclu dans la zone lumineuse du LHC, dont les dimensions sont de $15 \mu\text{m}$ dans le plan transverse et 5 cm le long du faisceau. La taille transverse est donc du même ordre que la résolution en paramètre d'impact à grand P_T qui est de l'ordre de $10 \mu\text{m}$. La détermination du vertex primaire est donc utile soit lorsqu'on doit la connaître avec une précision meilleure que la taille du faisceau (cas de l'étiquetage des b), soit lorsque on doit connaître sa position en z , cas de la reconstruction des photons en particulier dans le canal $H \rightarrow \gamma\gamma$ (voir section suivante). L'algorithme procédant à la reconstruction inclusive de vertex primaire commence par examiner le z des péricées de toutes les traces de faible paramètres d'impact, et à chercher les points sur l'axe du faisceau les plus proches d'un maximum de traces. Le vertex commun de ces traces proches sont déterminés, avec une petite itération pour supprimer les traces contribuant le plus au χ^2 . Le vertex principal et les vertex des événements de biais minimum (du moins ceux avec une multiplicité minimale) sont ainsi déterminés.

2.5.3 Développements récents et futurs

L'algorithme de tracking par défaut actuel procède à une recherche rapide de la position longitudinale du vertex primaire avant de trouver les traces proprement dit, afin de ne pas être pénalisé en rapidité à haute luminosité. Cela peut donner lieu à une perte d'efficacité de reconstruction de traces dans certains cas pathologiques, l'optimisation devra donc être revue à basse luminosité.

Les interactions hadroniques dans le trajectographe donnent lieu à des vertex de traces multiples. Ces traces peuvent ensuite polluer une analyse : la plupart auront un grand paramètre d'impact et/ou pas d'impact dans la b-couche de pixel, si bien qu'elles peuvent être nettoyées facilement, mais il en reste qui, par chance sont associés à un impact dans la b-couche et ont un paramètre d'impact. Reconstruire ces vertex de façon inclusive permettrait d'étiqueter ces traces comme mauvaises. D'autre part, cela permet une cartographie de la matière dans le détecteur plus précise que les conversions habituellement utilisés (même si les conversions sont sensibles à la longueur de radiation qui est la quantité vraiment intéressante pour l'étalonnage fin du calorimètre électromagnétique, alors que les interactions hadroniques sont sensibles à la longueur d'interaction).

La position de la zone lumineuse pourra varier au cours du temps. Même si ça ne devrait poser aucune difficulté, un algorithme pour «suivre» la position et la taille de la zone lumineuse, ainsi que son inclinaison, au cours du temps devra être mis au point.

2.6 Identification des électrons et photons

Les clusters électromagnétiques sont recherchés avec un algorithme de fenêtre glissante. La granularité de la deuxième couche du calorimètre électro-magnétique, où électrons et photons déposent la plupart de leur énergie est de 0.025×0.025 en η et ϕ . Les clusters rectangulaires sont définis en nombre de cellules. Leur dimension a été optimisée pour la meilleure reconstruction des électrons et photons isolés. La fenêtre glissante choisie est de 5×5 ; lorsque l'énergie comprise dans la fenêtre est maximale dans les deux directions η et ϕ , le barycentre du cluster 3×3 au centre de la fenêtre 5×5 définit la *graine*. Les clusters finaux sont des clusters rectangulaires définis autour de cette *graine* : pour les électrons les dimensions optimales sont de 3×7 , pour les photons de 3×5 dans le baril et 5×5 dans les bouchons. Ces dimensions ont été choisies comme un compromis pour capturer la plus grande part de l'énergie de la particule initiale tout en minimisant les effets de bruit électronique, d'empilement et d'événement sous-jacent. L'extension est généralement plus grande en ϕ pour pouvoir capturer les photons de Bremsstrahlung ainsi que les électrons provenant de conversion de photon (le champ magnétique étale les trajectoires en ϕ).

L'identification des photons et électrons de grand P_T dépend des répartitions en énergie longitudinale et latérale dans la gerbe. La granularité plus fine en η ($0.025/8$) de la première couche permet la réjection des jets avec un pion neutre énergétique. Une réjection typique de 10000 pour une efficacité supérieure à 80% est obtenue. Cela permet à la contribution des photons mal-identifiés de ne pas être dominante dans la recherche du boson de Higgs dans le canal $H \rightarrow \gamma\gamma$.

Pour l'identification des électrons, une trace pointant sur le cluster est recherché, avec une impulsion correspondant à celle du cluster. Le TRT permet une réjection additionnelle pouvant atteindre 100 pour une efficacité de 90% (mais avec une forte dépendance en pseudo-rapacité). La réjection totale atteint 100000 pour une efficacité supérieure à 70%. Des critères d'isola-

tion supplémentaires utilisant les traces où le calorimètre peuvent aussi être utilisés, mais leur utilisation dépend du canal choisi.

Un algorithme alternatif, particulièrement efficace à basse impulsion et pour les électrons non-isolés commence par les traces chargées reconstruites : les traces satisfaisant les critères minimaux d'identification dans le TRT sont extrapolés dans le calorimètre électromagnétique : un cluster est construit autour de leur impact, la cohérence avec l'hypothèse électron est ensuite vérifiée.

La mesure précise des objets électro-magnétique est perturbé par la présence de matière devant le calorimètre (entre un 0.3 et 1.5 longueur de radiation, avec un maximum autour de $\eta=1.5$ correspondant au passage de câbles et tuyaux de refroidissement du détecteur interne entre les bouchons et l'extrémité du baril). Une telle longueur de radiation signifie que la gerbe électro-magnétique peut commencer déjà dans le détecteur interne, et laisser une partie de son énergie en amont du calorimètre. Cet effet est corrigé en pondérant en fonction de l'énergie du pré-échantillonneur et de la première couche du calorimètre. De surcroît, la conversion d'un photon ou l'émission d'un photon de Bremsstrahlung dur à petit rayon peut, en raison du champ magnétique, causer une fuite d'énergie en dehors du cluster, causant une traîne vers les basses énergies. Une résolution typique de 1.5% à $P_T = 50$ GeV est finalement obtenue.

La zone lumineuse donnée par le LHC aura une longueur d'environ 5 cm RMS. La variation du z du vertex primaire affecte la résolution en pseudo-rapacité du calorimètre. La segmentation fine de la première couche du calorimètre électromagnétique permet une estimation de la pseudo-rapacité des photons avec une précision de l'ordre du centimètre sur l'estimation du z du vertex primaire (pour les électrons cette mesure est donnée par la trace chargée). Il suffirait que la la détermination du z du vertex original ait une précision inférieure à $100 \mu\text{m}$ pour que celle-ci ait une influence négligeable sur la précision de reconstruction du photon. Cet ordre de grandeur est l'ordre de grandeur de la résolution en paramètre d'impact longitudinal pour une seule trace, un vertex à deux traces a donc une précision largement suffisante. Toute la difficulté est dans le fait de distinguer le vertex principal des vertex d'événements de biais minimum : dans le cas de $H \rightarrow \gamma\gamma$ si l'impulsion du boson de H est faible, la multiplicité et l'impulsion des traces accompagnant le boson de Higgs peuvent ne pas être significativement différente d'événements de biais minimum.

2.6.1 Développements en cours et futurs

Pour les électrons, le développement d'une reconstruction de traces dédiée doit conduire à un ajustement global qui peut prendre en compte l'énergie du cluster électromagnétique associé, sachant que les propriétés du cluster peuvent être modifiés par le Bremsstrahlung : en général, les photons de Bremsstrahlung sont inclus dans le cluster (dont l'énergie et la position ne sont pas modifiés), mais avec un Bremsstrahlung plus important le ou les photons peuvent sortir progressivement du cluster, voire être inclus dans un cluster séparé. Les progrès sur la reconstruction et la mesure des traces électrons auront un impact direct que la qualité de l'identification des électrons. Il est aussi possible de combiner la mesure de l'énergie dans le calorimètre avec la mesure de l'impulsion dans le trajectographe, ce qui amènerait un gain pour les faibles p_T .

La méthode de fenêtre glissante utilisée actuellement doit pouvoir être améliorée en particulier pour les photons convertis ou électrons ayant subi un Bremsstrahlung. Une possibilité serait d'utiliser les clusters topologiques (décrits Section 2.8.1) avec le risque d'une perte de linéarité, ou bien une approche adaptative.

Les nouveaux développements au niveau tracking permettent d'améliorer la reconstruction des conversions, voire rendent possible la reconstruction des conversions tardives, c.à.d dans les dernières couches de silicium voire même dans le TRT seul, cette information doit être utilisée pour :

- optimiser la réjection des conversions de photon asymétrique simulant un électron
- améliorer l'étalonnage des photons convertis, car pour ceux-là une partie de l'énergie peut s'échapper du cluster
- améliorer la séparation entre photon et électron
- améliorer la détermination du z du vertex primaire pour les événements $H \rightarrow \gamma\gamma$

La quantité de matière devant le calorimètre ayant significativement augmenté ces dernières années, une réoptimisation de l'identification (pour photons et électrons) est nécessaire. Il s'agit non seulement de réoptimiser les coupures (ou de tenter des combinaisons optimales de variables par réseau de neurone ou méthode similaire), mais aussi de concevoir de nouvelles variables qui soient plus robustes, pour regagner en efficacité.

2.7 Identification et reconstruction des muons

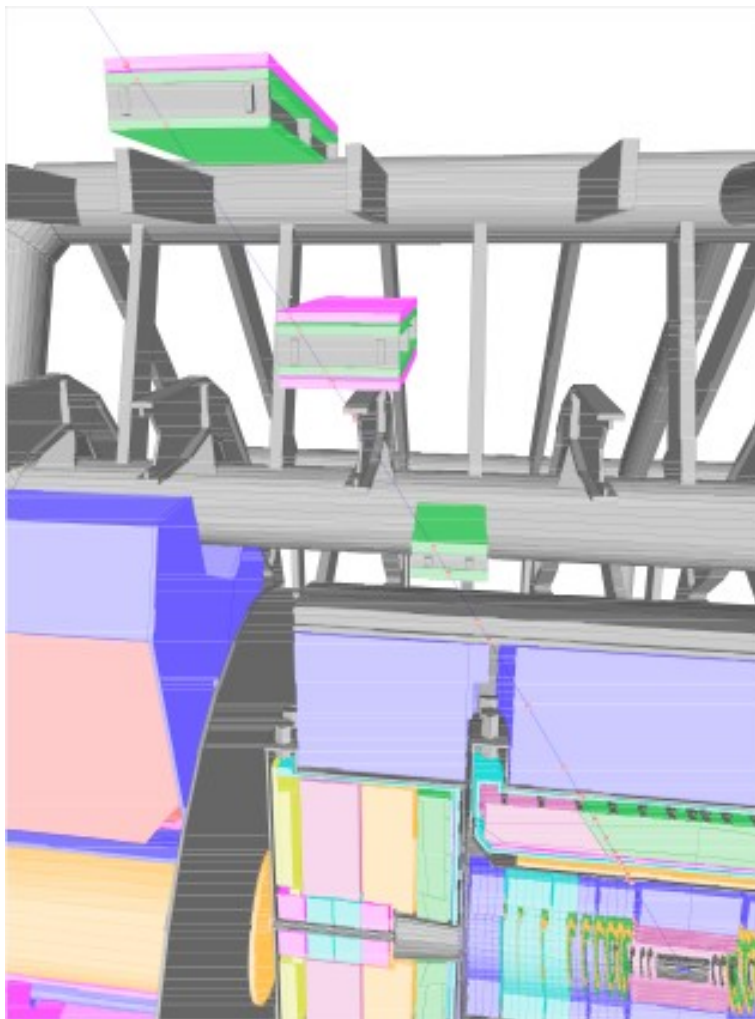


FIGURE 2.17: Trajectoire d'un muon depuis le centre d'Atlas jusque dans le spectromètre à muon. Seules les chambres à muon contribuant à la détection du muon considéré sont dessinées.

Le spectromètre à muons fournit par lui-même l'identification des muons et la reconstruction par la mesure dans les toroïdes (à l'aide de détecteurs de quatre technologies différentes) de typiquement trois segments de la trajectoire dans trois stations, à l'entrée, au milieu et à la sortie du spectromètre (voir Figure 2.17). L'efficacité est typiquement de 95%, l'inefficacité étant principalement due aux différents trous d'acceptance pour le passages des supports du détecteurs et des services. L'efficacité diminue à très grand P_T (au-delà de 500 GeV) en raison des pertes d'énergie catastrophiques, pour lesquels une gerbe électromagnétique accompagne le muon et perturbe la reconstruction de la trajectoire(voir Figure 2.18). En dessous de 6 GeV , la perte d'énergie du muon dans le calorimètre est de l'ordre de son énergie si bien que le muon peut émerger du calorimètre avec une impulsion inférieure au GeV de sorte qu'il ne traverse plus les trois stations.

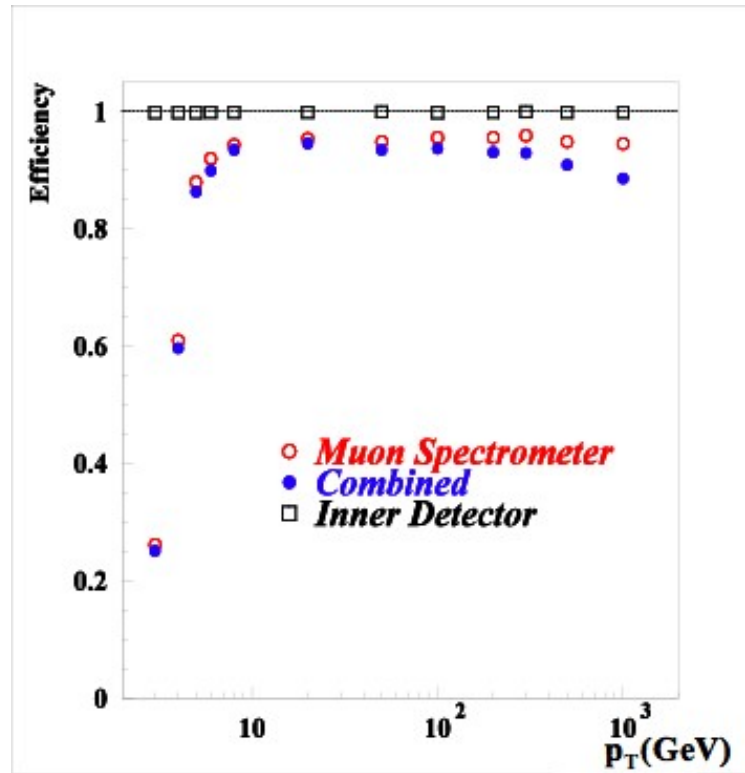


FIGURE 2.18: Efficacité de reconstruction d'un muon en fonction de son impulsion transverse avec le trajectographe interne seul ("Inner Detector", avec le spectromètre à muon seul ("Muon Spectrometer"), ou combiné ("Combined")

Le muon identifié est propagé vers le point d'interaction à travers le calorimètre, en tenant compte de la perte d'énergie et de la diffusion multiple, puis est combiné avec une trace du détecteur interne, ce qui permet d'améliorer la résolution en P_T pour les impulsions inférieures à 20 GeV . Deux méthodes sont étudiées pour combiner le segment de trace dans le trajectographe avec le segment de trace dans le spectromètre à muon : la combinaison statistique des paramètres des deux segments à l'aide des matrices de covariances complètes, ou bien le réajustement complet d'une seule trajectoire passant par les points associés aux deux segments. La première est plus robuste, la seconde est plus délicate à mettre au point car elle exige un traitement correct des erreurs d'extrapolation et sur les points sur une trajectoire de plus de 10 m, par contre elle ouvre la possibilité de refaire a posteriori un pattern recognition local,

c-à-d par exemple abandonner un point dans les muons qui deviendrait clairement en dehors de la trajectoire contrainte par le trajectographe interne.

2.7.1 Développements en cours et futurs

L'énergie perdue dans le calorimètre peut être paramétrisée, elle peut aussi être mesurée, mais au risque d'une contamination par d'autres particules. L'optimisation en est délicate.

Pour identifier les muons de faible impulsion (en dessous de 6 GeV), une approche alternative a été développée récemment. Les traces du détecteur interne sont extrapolées et associées aux segments qui n'auront pas été déjà assignés à un muon complet.

2.8 Reconstruction des jets

La reconstruction des jets est faite soit avec un algorithme de cône, soit avec un algorithme Kt dont l'utilisation dans les collisionneurs hadroniques est peu courante. Les cellules sont sommées en tours de granularité $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$, qui correspond à la granularité des calorimètres hadroniques. Les tours qui auraient une énergie totale négative (en raison d'une fluctuation du bruit) sont sommées avec leurs voisines, car une énergie négative ne permet pas de calculer le critère de proximité Kt. Pour optimiser la résolution en énergie, des poids sont appliqués qui tiennent compte de la segmentation longitudinale du calorimètre. La meilleure résolution est obtenue en utilisant un ensemble de sept poids de la forme

$$w_i(|\eta|) = a_i(|\eta|) + b_i(|\eta|)/E + c_i(|\eta|) \log(E),$$

où l'indice court sur les couches des calorimètres électromagnétiques et hadroniques. La dépendance en $|\eta|$ permet de prendre en compte la profondeur effective variable (en terme de longueur de radiation) des calorimètres, la distribution de la matière devant le calorimètre et la performance dégradée dans la transition baril-bouchon. Une résolution typique $\sigma_E/E = 65\%/\sqrt{(E)} \oplus 2\%$ est obtenue à faible pseudo-rapidité $|\eta| \leq 0.8$.

2.8.1 Développements en cours et futurs

Un développement récent est de construire les jets non plus à partir des tours mais à partir des clusters topologiques. L'algorithme de cluster topologique associe une cellule «graine» d'énergie significative (supérieure à a fois le bruit attendu) à ses voisines, tant que l'énergie de ses voisines est supérieure à un seuil (b fois le bruit attendu). Puis un halo de cellules dont l'énergie est supérieure à un seuil c fois le bruit attendu est ajouté au cluster. Finalement, une deuxième itération, recherche les maxima locaux dans chaque cluster pour éventuellement les diviser. Ces trois seuils a , b et c sont optimisés pour que chaque cluster corresponde au dépôt d'énergie d'une particule, ou d'un jet compact. Des valeurs élevées favorise de petit clusters, mais posent un problème de linéarité, en raison d'une perte de signal sur les bords, tandis que des valeurs faibles favorise la contribution du bruit. Les valeurs optimales sont 4, 2 et 0. Chaque cluster est ensuite calibré séparément, après une classification des clusters à dominantes électromagnétiques. Ces clusters sont ensuite rassemblés en jets avec les algorithmes classiques, dont la résolution devrait être meilleure que pour les jets faits à partir des tours.

D'autre part, les tours ne peuvent pas être écrites dans les AOD étant donné leur nombre (près de 6000), alors qu'il est possible d'écrire les quelques centaines de clusters dans les AOD. Cela permet éventuellement de réoptimiser depuis les AOD la reconstruction des jets d'une façon spécifique à une analyse, ou par exemple d'accéder à une sous structure des jets.

Il faut mentionner que l'utilisation des traces dans la reconstruction de l'énergie des jets a été tentée (ce type d'algorithme était très performant au LEP) mais n'a pour l'instant pas donné de résultats convaincants, la difficulté étant d'améliorer la résolution sans ajouter de traîne supplémentaire, sachant qu'en raison de la quantité de matière dans le trajectographe l'efficacité de reconstruction des traces chargées des hadrons est inférieure à 95%.

2.9 Energie transverse manquante

La reconstruction de l'énergie transverse manquante est essentielle pour l'identification et la reconstruction des canaux avec un neutrino ou toute autre particule hypothétique invisible (le LSP de SuSy par exemple). L'algorithme de référence actuel calcule l'impulsion transverse visible dans le calorimètre en sommant les impulsions de toutes les cellules au-dessus d'un seuil défini en nombre du sigma du bruit attendu. La résolution trouvée est proportionnelle à la racine carrée de l'énergie transverse visible

$$\sigma_{P_x} = 0.5 \sqrt{\sum E_T}$$

et est relativement indépendante du canal de physique (e.g $W \rightarrow \ell \nu_\ell$ ou $A^0 \rightarrow \tau^+ \tau^-$).

L'impulsion transverse manquante peut être utilisée pour reconstruire les masses de particules avec un ou plusieurs neutrinos dans l'état final. Un exemple typique est $A^0 \rightarrow \tau^+ \tau^-$ (ou A est le boson de Higgs pseudoscalaire du MSMM), où on peut supposer que chaque neutrino est colinéaire aux produits de désintégration de chaque τ , ce qui conduit à un système d'équations suffisamment contraint.

2.9.1 Développements en cours et futurs

L'algorithme de référence a été amélioré progressivement en tenant compte des muons et des électrons identifiés, et des jets reconstruits. La possibilité de reconstruire l'énergie transverse à partir des même clusters topologiques décrits précédemment est également à l'étude.

2.10 Etiquetage des b-jet

L'étiquetage des b-jet est importante pour la recherche d'un boson de Higgs léger (dans le mode de production associée $t\bar{t}H(b\bar{b})$), mais aussi pour identifier la longue liste de désintégration où des quarks b apparaissent. Le principe de base est que dans un jet de 50 GeV, les quelques 5 traces (en moyenne) issues de la désintégration du hadron B, proviennent d'un vertex détaché du vertex primaire de 5mm (en moyenne, avec une distribution exponentielle). Ces traces auront un paramètre d'impact significativement positif alors que les traces issues du vertex primaire auront un paramètre d'impact moyen nul. Un premier algorithme pour l'étiquetage des b-jet repose sur la combinaison des paramètres d'impact des traces chargées, après application de critères de qualité sévères pour éviter les traces faussement déplacées. Une efficacité de 60% peut être obtenue, avec une réjection typique des jets légers d'un facteur 100 pour $H \rightarrow b\bar{b}$ avec $m_H = 120$ GeV. Cependant la réjection dépend fortement de la pseudo-rapacité et de l'énergie transverse du jet. La réjection est dégradée à grande pseudo-rapacité principalement en raison de la matière dans le trajectographe qui détériore la résolution en paramètre d'impact, et surtout crée des traces de grands paramètres d'impact provenant de conversions et d'interactions hadroniques. La réjection est optimale pour les jets de P_T entre 50 et 150 GeV. A plus faible impulsion les traces issus du hadron B (dont le P_T est de quelques GeV) souffrent d'un paramètre d'impact dégradé par la diffusion multiple, voire ne sont pas reconstruites du tout. A plus grande

impulsion, trois effets se conjuguent pour dégrader la réjection, malgré l'amélioration de la résolution en paramètre d'impact :

- il y a une plus grande proportion de traces venant du vertex primaire
- il y a plus de traces d'impulsion notables venant d'interactions hadroniques ou de conversions
- les traces du B peuvent être mal séparées car plus colimatées.

Un algorithme plus sophistiqué a été développé qui consiste à reconstruire en trois dimensions le vertex primaire (en supposant la connaissance de la position du faisceau qui est une contrainte très utile en raison de sa largeur transverse de $15 \mu\text{m}$), et à reconstruire en trois dimensions de façon inclusive le vertex secondaire du B. Cet algorithme permet une réjection jusqu'à cinq fois meilleure car :

- la reconstruction en trois dimensions des vertex (même si la résolution du tracking en z , c.à.d suivant l'axe du faisceau est près de 10 fois moins bonne que dans le plan transverse)
- la reconstruction d'un vertex secondaire est plus robuste envers les traces mal reconstruites de grand paramètre d'impact : auparavant toute trace de grand paramètre d'impact était délétère, mais maintenant il faut aussi qu'elle passe par chance près d'une autre trace pour former un vertex séparé
- la masse invariante des traces associés au vertex secondaire est un critère de réjection utile contre les jets de charme

Toutefois les dépendances en η et P_T sont similaires.

2.10.1 Développements en cours et futurs

Les développements en cours portent principalement sur la réoptimisation des algorithmes, sur la stratégie à adopter au démarrage, lorsque le trajectographe ne sera pas encore aligné, ainsi que sur l'étalonnage des efficacités et réjections sur les données.

2.11 Identification des τ

L'identification des désintégration hadroniques des τ est importante pour de nombreux canaux de désintégration où le τ intervient, par exemple pour le Higgs du modèle standard dans la production Vector-Boson-Fusion $qqH(\tau^+\tau^-)$ ou pour le canal de découverte du boson de Higgs supersymétrique pseudo-scalaire $A^0 \rightarrow \tau^+\tau^-$. Le τ étant le boson le plus lourd apparaît aussi souvent dans les désintégrations d'autre particules supersymétriques. Les désintégrations hadroniques du τ apparaissent comme des jets isolés très fins (la finesse du jet peut être mesurée grâce à la granularité fine du calorimètre électro-magnétique) associé à une faible multiplicité de trace rapportée à l'énergie du jet. Les traces auront tendance à avoir un paramètre d'impact positif mais cet effet est bcp moins net que pour les hadrons beaux, en moyenne $90 \mu\text{m}$ contre $500 \mu\text{m}$. Les variables calorimétriques et concernant les traces doivent ensuite être combinées de manière optimale. La dépendance en P_T de la qualité d'identification des τ est très importante, car la taille de la gerbe varie rapidement, ainsi que la multiplicité pour les jets QCD. Le point de fonctionnement (en terme d'efficacité/réjection) doit être choisi en fonction de l'analyse effectuée. Par exemple, à 50 GeV, on peut obtenir une réjection de 1000 pour une efficacité de 30% ou de 100 pour une efficacité de 60%.

3 RECHERCHE DU PETIT HIGGS DANS ATLAS

Le «Petit Higgs» («Little Higgs») est une classe de modèles effectifs relativement récents (2001) qui règlent le problème de hiérarchie et de divergences quadratiques du modèle standard en introduisant de nouvelles particules (de même statistique, contrairement à la supersymétrie) de masse de l'ordre du TeV, le boson de Higgs restant relativement léger avec des propriétés standard, et les paramètres pouvant être choisis pour échapper aux contraintes de toutes les observables actuelles. Parmi les différentes variantes, le modèle du «Littlest Higgs» [16]⁷ prédit trois bosons de jauge lourds, W_H , Z_H (dégénérés en masse) et A_H , un top lourd T et un nouveau secteur du Higgs lourd noté ϕ^0 , ϕ^+ , ϕ^{++} .

La possibilité de détecter les nouvelles particules prédites par le modèle du Petit Higgs au LHC a été étudiée. Nous aborderons les canaux mettant en jeu le nouveau quark lourd T , puis les nouveaux bosons de jauge A_H , W_H et Z_H (de façon plus détaillée car analysés en partie au LAL), et finalement le boson de Higgs doublement chargé ϕ^{++} . Les résultats présentés sont extraits de [18–20], complétées par des études plus récentes dans [9, 21, 22].

Les rapports d'embranchement du boson de Higgs sont pris identiques au modèle standard. La simulation rapide de Atlas Atlfast a été utilisée. Atlfast fournit une réponse paramétrée du détecteur aux jets, électrons, muons, photons isolés, et énergie manquante transverse. Les paramétrisations d'Atlfast ont été dérivées d'un grand nombre d'étude sur la simulation complète du détecteur, à haute luminosité ($10^{34} \text{ cm}^{-2} \text{ s}^{-1}$). Les jets sont reconstruits dans un cône de taille $\Delta R=0.4$. Les résultats seront donnés pour une luminosité intégrée de 300 fb^{-1} , correspondant aux données collectées pendant trois ans à haute luminosité. Figure 3.19 montre que le boson de Higgs standard sera déjà découvert avec une luminosité de 30 fb^{-1} , donc l'hypothèse est faite que la masse du boson de Higgs sera connue. Tous les signaux recherchés comprennent un ou plusieurs leptons ou photons isolés de grande impulsion transverse dans l'état final, donc passeront le déclenchement d'Atlas avec une forte probabilité, ce point n'a donc pas été étudié en détail. La plupart des bruits de fond a été simulé avec PYTHIA [23]. Des générateurs dédiés ont été utilisés dans certains cas.

3.1 Recherche du quark lourd T

Le quark lourd T peut être produit par deux mécanismes :

- production QCD par les processus $gg \rightarrow T\bar{T}$ and $q\bar{q} \rightarrow T\bar{T}$ qui ne dépendent que de la masse de T
- production par échange de W $qb \rightarrow q'T$ conduisant à un seul quark T dans l'état final, et donc avec une chute de la section efficace plus lente lorsque M_T augmente. Ce processus dépend des paramètres du modèle, et, en particulier, de l'angle de mélange du T avec le quark top du modèle standard.

Les couplages de Yukawa de T sont donnés par deux constantes λ_1 and λ_2 [16]. L'état propre de masse physique est un mélange de t et T , et les différents couplages dépendent de λ_1 , λ_2 et f qui déterminent le mélange de t , T et leurs masses. Deux de ces paramètres peuvent être réinterprétés comme les masses du top et du T . Le troisième peut être choisi comme λ_1/λ_2 . Celui-ci détermine les mélanges et donc le couplage TbW qui contrôle le taux de production par le processus $qb \rightarrow q'T$.

7. Plusieurs erreurs de calcul dans [16] ont été relevées par [17]. Elles n'affectent toutefois aucun des résultats présentés.

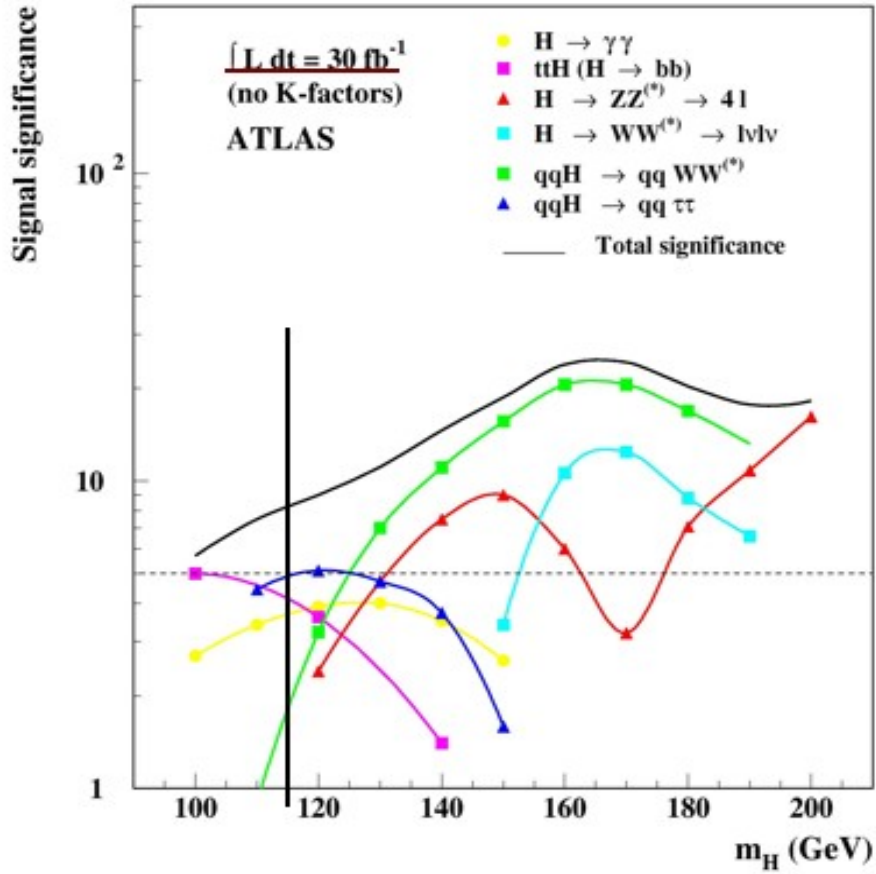


FIGURE 3.19: Visibilité du boson de Higgs dans Atlas avec une luminosité de 30 fb^{-1} [24]. La zone non représentée avec $m_H \geq 200 \text{ GeV}$ est également couverte à plus de 5σ .

Les calculs donnés par [16] indiquent que la production simple domine pour une masse de T supérieure à 700 GeV. Comme nous pensons être sensible à des masses supérieures à celles-ci, seule la production simple sera considérée dans la suite. Nous supposons une section efficace $\sigma = 200 \text{ fb}$ pour $M_T = 1.0 \text{ TeV}$ et $\lambda_1/\lambda_2 = 1$. Les événements générés avec PYTHIA seront normalisés à ces valeurs. Les taux de désintégrations sont les suivants :

$$\Gamma(T \rightarrow tZ) = \Gamma(T \rightarrow tH) = \frac{1}{2}\Gamma(T \rightarrow bW) = \frac{\kappa^2}{32\pi}M_T \quad (3.1)$$

avec $\kappa = \lambda_1^2 / \sqrt{\lambda_1^2 + \lambda_2^2}$ impliquant que T est une résonance étroite. La dernière de ces désintégrations serait attendue pour un quark de quatrième génération de charge $2/3$; alors que les deux premières, étant des courants neutres changeant la saveur, sont plus caractéristiques du modèle du "petit Higgs". La mise en évidence de ces canaux est décrite dans la suite.

3.1.1 Etude de la désintégration $T \rightarrow Zt$

Ce canal peut être observé dans l'état final $Zt \rightarrow \ell^+\ell^-\ell\nu b$, pour lequel l'événement contient trois leptons isolés, dont deux formant la masse du Z , un b -jet et de l'énergie manquante. Le bruit de fond est dominé par WZ , ZZ and tbZ . Les événements sont sélectionnés comme suit :

- trois leptons isolés (e ou μ) avec $p_T > 40$ GeV et $|\eta| < 2.5$, dont un lepton avec $p_T > 100$ GeV.
- pas d'autres leptons avec $p_T > 15$ GeV.
- $E_T^{miss} > 100$ GeV.
- au moins un b -jet étiqueté

La présence de leptons est suffisante pour le déclenchement. Une paire de leptons de même saveur et de charges opposées doit être compatible avec la masse du Z à 10 GeV près. Le troisième lepton est supposé venir d'un W dont l'impulsion est reconstruite en s'aidant de l'énergie manquante. L'efficacité de sélection est 3.3% pour $M_T = 1$ TeV. La masse invariante du système Zt peut enfin être reconstruite en incluant le b -jet.

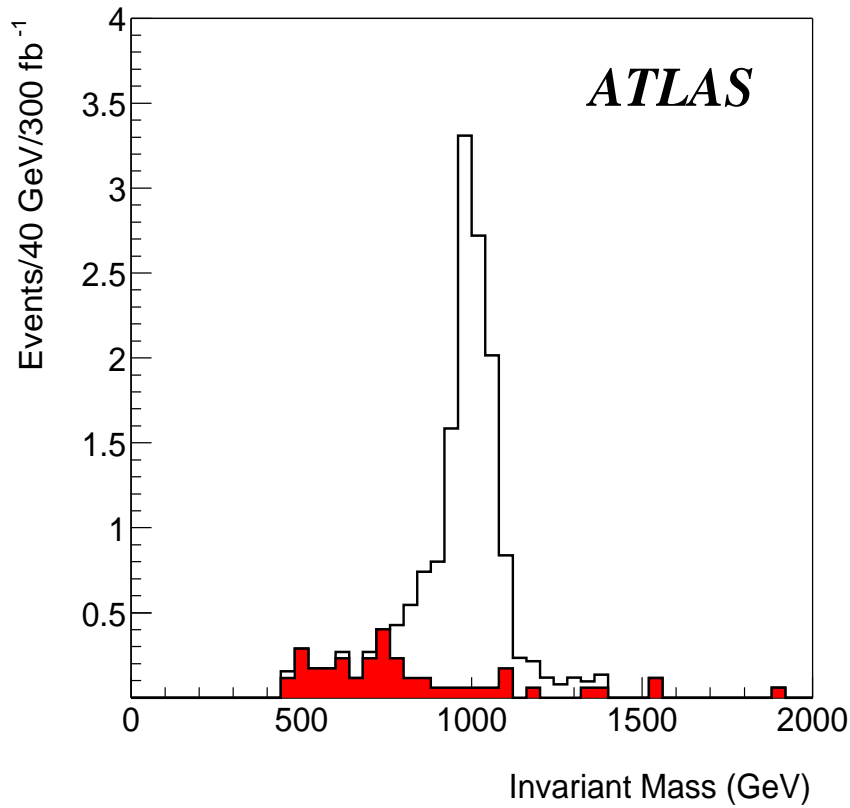


FIGURE 3.20: Le signal $T \rightarrow Zt$ est montré pour une masse de 1 TeV. Le bruit de fond est dominé par les événements WZ and tbZ .

Figure 3.20 indique la masse reconstruite pour $M_T = 1$ TeV avec un signal clairement visible au-dessus du bruit de fond. Le bruit de fond est largement dominé par la production du modèle standard de tbZ . Le potentiel de découverte peut être estimé. En demandant un pic de signification supérieure à 5σ contenant au moins 10 événements reconstruits implique que pour $\lambda_1/\lambda_2 = 1$ (resp. 2) et une luminosité intégrée de 300 fb^{-1} , le quark lourd T est observable jusqu'à une masse de $M_T < 1050$ (resp. 1400) GeV. A ces masses, la production de T est effectivement dominée par la production simple, justifiant a posteriori d'avoir négligé la production de paire $T\bar{T}$.

3.1.2 Etude de la désintégration $T \rightarrow Wb$

Ce canal peut être reconstruit dans le canal $\ell\nu b$. La sélection suivante est appliquée.

- au moins un lepton avec $p_T > 100$ GeV.
- un seul b -jet avec $p_T > 200$ GeV.
- pas plus de 2 jets avec $p_T > 30$ GeV.
- la masse de la paire de jets de plus grand p_T est supérieure à 200 GeV.
- $E_T^{miss} > 100$ GeV.

Le lepton donne le déclenchement. Le bruit de fond vient de $t\bar{t}$, production simple de top et production QCD de $Wb\bar{b}$. La contrainte d'avoir un seul b -jet et au moins un lepton de très grand p_T est efficace pour réduire ce bruit de fond. La contrainte d'avoir au plus deux jets énergétique est efficace contre le bruit de fond $t\bar{t}$. L'efficacité de sélection est 14% pour $M_T = 1$ TeV. Le rapport signal sur bruit est moins bon que dans le cas précédent en raison de la contribution $t\bar{t}$. Le potentiel de découverte peut être évalué au delà de 5σ pour $M_T < 2000$ (resp. 2500) GeV, pour $\lambda_1/\lambda_2 = 1$ (resp 2).

3.1.3 Etude de la désintégration $T \rightarrow Ht$

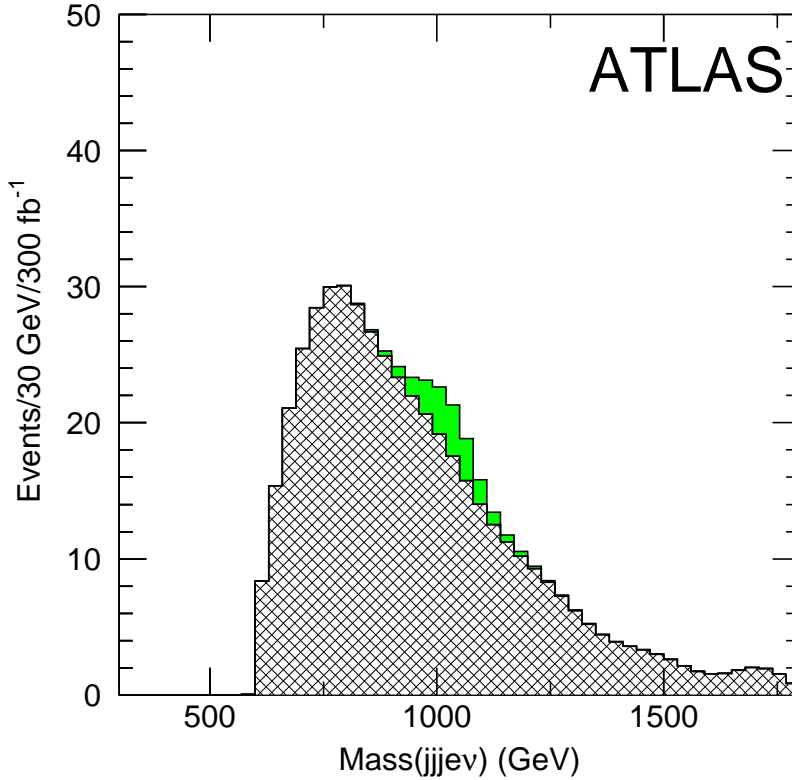


FIGURE 3.21: Le signal $T \rightarrow Ht$ est montré pour une masse de 1 TeV. Le bruit de fond est dominé par les événements $t\bar{t}$.

La topologie finale dépend de la masse du Higgs. Seul a été étudié le cas $m_H = 120$ GeV pour lequel la désintégration $b\bar{b}$ domine. La désintégration semileptonique du top $t \rightarrow Wb \rightarrow \ell\nu b$ donne un lepton favorable au déclenchement. La sélection procède ainsi :

- un lepton isolé e or μ avec $p_T > 100$ GeV et $|\eta| < 2.5$.
- trois jets avec $p_T > 130$ GeV.
- au moins un b -jet étiqueté.

De plus, au moins une paire de jets comportant le b -jet doit avoir une masse entre 110 et 130 GeV (rappelons que la masse du Higgs est supposée connue). L'événement est rejeté si une paire de jets a une masse entre 70 et 90 GeV, pour diminuer le bruit de fond $t\bar{t}$ comportant souvent un W hadronique. L'énergie transverse et le lepton sont combinés pour tester leur cohérence avec la désintégration $W \rightarrow \ell\nu$, et évaluer l'impulsion du W . L'efficacité de sélection est 2.3% pour $M_T = 1$ TeV. La masse invariante du W reconstruit, du H et d'un jet supplémentaire est calculée, voir Figure 3.21. Le bruit de fond est dominé par les événements $t\bar{t}$. La largeur de la résonance T est dominée par la résolution expérimentale. La signification est moins bonne que pour les autres canaux. Pour $\lambda_1/\lambda_2 = 1$, elle est de 4σ for $M_T = 1$ TeV, tombant à 3σ for $M_T = 700$ GeV.

3.2 Recherche de nouveaux bosons de jauge

Le modèle prédit de nouveaux bosons de jauge lourds : deux chargés W_H^\pm et deux neutres Z_H and A_H . Les bosons W_H and Z_H sont quasi-dégénérés en masse et normalement plus lourd que A_H . Par des arguments de réglage fin, une limite supérieure peut être donnée :

$$M_{W_H, Z_H} < 6 \text{ TeV} (m_H / 200 \text{ GeV})^2,$$

soit 2 TeV pour $m_H = 120 \text{ GeV}$ et 6 TeV pour $m_H = 200 \text{ GeV}$.

Deux nouveaux couplages sont définis. Ils peuvent être choisis comme étant deux angles θ et θ' . Une fois les masses des nouveaux bosons spécifiées θ détermine les couplages du Z_H et θ' ceux de A_H . Ces trois bosons seront découverts par leur désintégrations leptoniques pour lesquels l'efficacité de détection est importante. Toutefois, pour les distinguer des bosons lourds d'autres modèles, les désintégrations caractéristiques $Z_H \rightarrow ZH$ et $W_H \rightarrow WH$ doivent être observées [25]. L'étude de l'observabilité de ces états finaux au LHC (quel que soit le modèle) n'avait pas été faite avant ces travaux réalisés en collaboration entre le LAL et l'université de Valence [18, 19].

3.2.1 Section efficace et rapports d'embranchement

Les largeurs partielles de désintégration des bosons vecteurs lourds W_H et Z_H notés V_H , en fermions sont

$$\Gamma(V_H \rightarrow f\bar{f}') = \frac{C}{12\pi} (g_V^2 + g_A^2) M_{V_H} \quad (3.2)$$

avec C le facteur de couleur, et g_V et g_A les couplages vecteur et axial-vecteur, définis par :

$$g_V = -g_A = \begin{cases} \frac{g}{2} \cot \theta T_3 & \text{pour } Z_H \\ \frac{g}{2\sqrt{2}} \cot \theta & \text{pour } W_H \end{cases} \quad (3.3)$$

(avec g la constante de couplage et T_3 l'isospin faible soit $\pm 1/2$). Auront été négligées les corrections d'espace de phase qui sont d'au plus 10% pour le cas le plus défavorable $Z_H \rightarrow t\bar{t}$ si $M_{Z_H} = 1 \text{ TeV}$. La largeur partielle en une seule paire de fermions est donc double pour W_H par rapport à Z_H , mais le W_H peut donner 12 paires de fermions différentes (en incluant les couleurs), contre 24 pour Z_H , si bien que les largeurs totales en fermions sont identiques :

$$\Sigma_{\text{fermions}} \Gamma(V_H \rightarrow f\bar{f}') = \Sigma_{\text{fermions}} \Gamma(Z_H \rightarrow f\bar{f}') = \frac{g^2 \cot^2 \theta}{4\pi} M_{V_H} \quad (3.4)$$

Les largeurs partielles de désintégration en bosons de jauge du Modèle Standard :

$$\Gamma(V_H \rightarrow VH) = \Gamma(Z_H \rightarrow WW) = \Gamma(W_H \rightarrow WZ) = \frac{g^2 \cot^2 2\theta}{192\pi} M_{V_H} \quad (3.5)$$

Ces largeurs sont identiques entre W_H et Z_H . La largeur totale est :

$$\Gamma/M = (3.4 \cot^2 \theta + 0.071 \cot^2 2\theta) \% \quad (3.6)$$

Les rapports d'embranchement ne dépendent donc que de $\cot \theta$, et en particulier ne dépendent pas de $M_{Z_H} = M_{W_H}$ ni de M_H . Les rapports d'embranchement vérifient les relations simples suivantes pour les désintégrations en bosons :

$$B(Z_H \rightarrow ZH) = B(W_H \rightarrow WH) = B(Z_H \rightarrow WW) = B(W_H \rightarrow WZ) \quad (3.7)$$

et pour les désintégrations en fermions :

$$B(Z_H \rightarrow ll) = 1/2 B(W_H \rightarrow l\nu) \quad (3.8)$$

La Figure 3.22 donne le comportement des rapports d'embranchement en fonction de $\cot \theta$. Les modes bosoniques dominent à petit $\cot \theta$, sont totalement supprimés à $\cot \theta = 1$, restent faible mais non nuls pour $\cot \theta > 1$. Les désintégrations fermioniques, nulles à petit $\cot \theta$ augmentent avec $\cot \theta$, s'approchant pour $\cot \theta > 0.5$, d'une valeur asymptotique de 4% pour Z_H (e.g. $Z_H \rightarrow e^+e^-$) et 8% pour W_H (e.g. $W_H \rightarrow e^-\nu$).

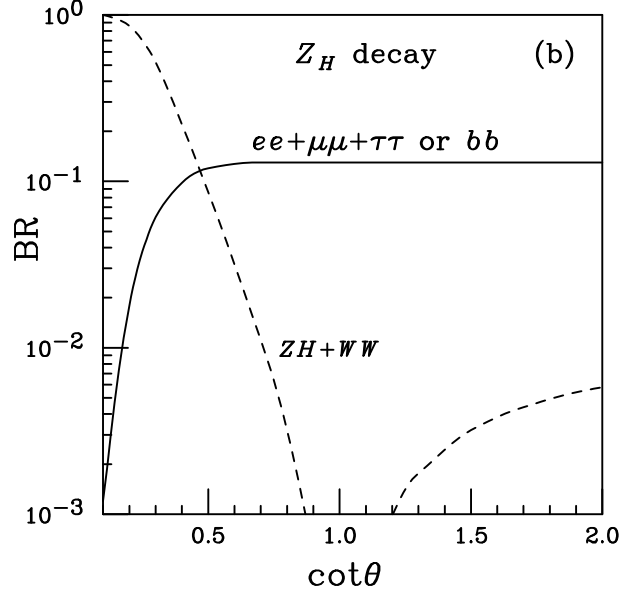


FIGURE 3.22: Rapports d'embranchement du boson de jauge lourd Z_H en fonction de $\cot \theta$ [16].

Au LHC, les bosons lourds sont produits principalement par les diagrammes $u\bar{u}/d\bar{d} \rightarrow Z_H$ et $u\bar{d}/d\bar{u} \rightarrow W_H$, leur section efficace de production (voir Figure 3.23) est donc liée à $\Gamma(V_H \rightarrow f\bar{f}')$: elle est proportionnelle à $\cot^2 \theta$ et double pour W_H par rapport à Z_H , voir Tableau 3.2.

La dépendance en $\cot \theta$ de la section efficace de production des V_H multipliée par le rapport d'embranchement de $V_H \rightarrow ZH$ est tracée sur la Figure 3.24. La valeur $\cot \theta = 0.5$ sera choisie comme référence, n'étant pas particulièrement favorable, ni défavorable. Les principaux rapports d'embranchement pour $\cot \theta = 0.5$ sont indiqués dans le Tableau 3.1 La section efficace correspondante est indiquée pour différentes masses dans le Tableau 3.2.

mode	$B(Z_H)$	$B(W_H)$ (%)
$V_H \rightarrow VH$	4,3%	4,3%
$V_H \rightarrow \ell\nu/\ell$	3,8%	7,6%

TABLE 3.1: Rapports d'embranchement pour les principaux mode de désintégrations de W_H et Z_H calculés pour $\cot \theta = 0.5$.

Les états finaux utilisables dépendent des rapports d'embranchement du boson de Higgs donc de sa masse (voir Figure 3.25). L'analyse a été faite pour $m_H = 120$ GeV et $m_H =$

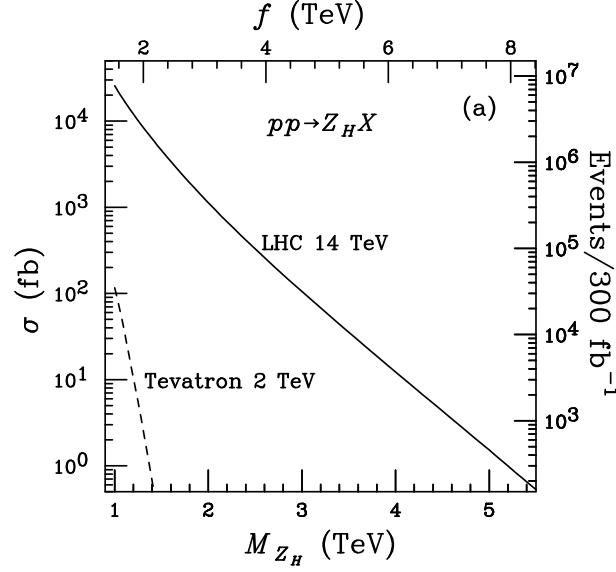


FIGURE 3.23: Section efficace de production du boson de jauge lourd Z_H , en fonction de sa masse et de f , pour $\cot \theta = 1$ [16]. La section efficace de production de W_H est double.

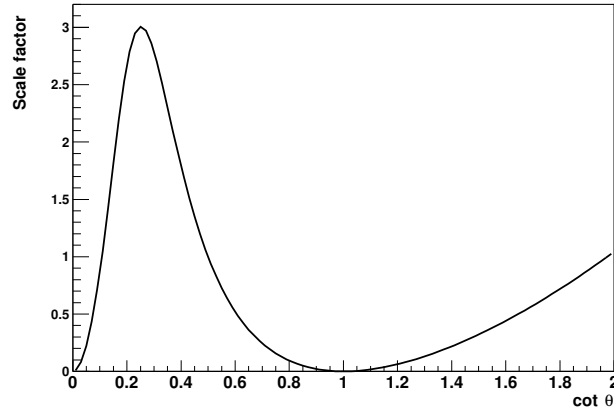


FIGURE 3.24: Dépendance en $\cot \theta$ de $\sigma.B(V_H \rightarrow V + H)$.

200 GeV correspondant à des scénarios complètement différents. Les rapports d'embranchement du Higgs notables pour ces masses sont indiqués Tableau 3.3.

Pour opérer le choix des états finaux satisfaisant la double condition d'avoir une section efficace suffisante et une bonne signature, les courbes représentées sur les Figures 3.26 pour Z_H et 3.27 pour W_H sont utiles. Elles indiquent, pour $M_{V_H} = 1$ TeV et en fonction de la masse du boson de Higgs, la section efficace multipliée par le rapport d'embranchement des différents états finaux. Ces états ont été choisis pour leur signature, ayant au moins 2 leptons, 2 quarks b ou 2 photons. Une partie a été étudié au LAL, l'autre à l'université de Valence, les canaux suivants ont été choisis :

masse (GeV)	σ_{Z_H} (fb)	σ_{W_H} (fb)
850	9500	19000
1000	4750	9500
1250	2375	4750
1500	950	1900
1750	475	950
2000	238	475

TABLE 3.2: Section efficace de production de Z_H et W_H , pour $\cot \theta = 0.5$.

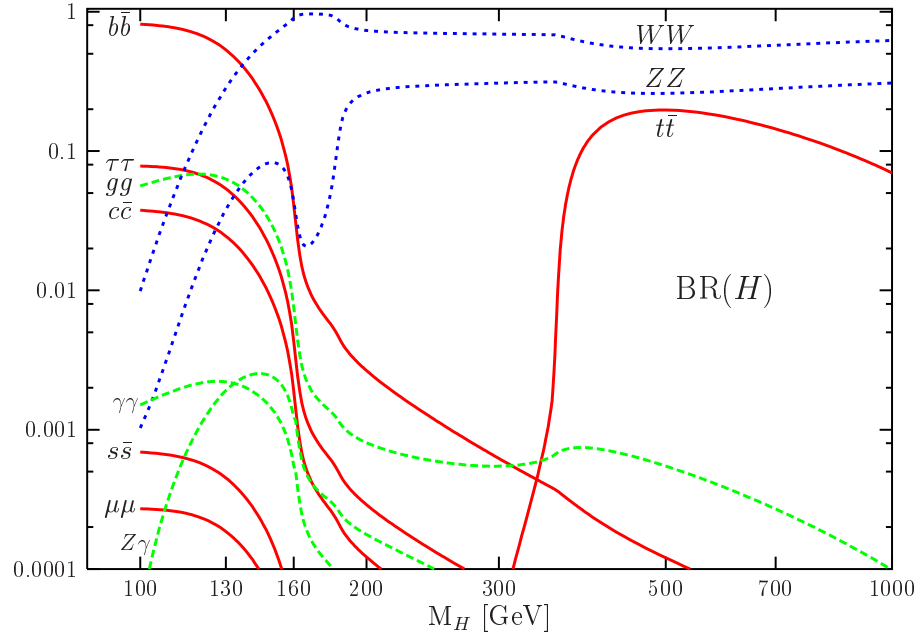


FIGURE 3.25: Rapport d'embranchement du boson de Higgs standard en fonction de sa masse [26].

- pour $M_H = 120$ GeV :

$$\begin{aligned}
 & \left. \begin{aligned} 1) \quad Z_H &\rightarrow Z H \rightarrow q\bar{q} \gamma\gamma \\ 2) \quad W_H &\rightarrow W H \rightarrow qq' \gamma\gamma \end{aligned} \right\} \rightarrow \text{LAL} \\
 & \left. \begin{aligned} 3) \quad Z_H &\rightarrow Z H \rightarrow l^+ l^- b\bar{b} \\ 4) \quad W_H^\pm &\rightarrow W^\pm H \rightarrow l^\pm \nu b\bar{b} \end{aligned} \right\} \rightarrow \text{Valence}
 \end{aligned} \tag{3.9}$$

- pour $M_H = 200$ GeV :

$$\begin{aligned}
 & \left. \begin{aligned} A1 \quad Z_H &\rightarrow Z H \rightarrow l^+ l^- W^+ W^- \rightarrow l^+ l^- l^+ \nu l^- \nu \\ A2 \quad W_H &\rightarrow W H \rightarrow l \nu W^+ W^- \rightarrow l \nu l^+ \nu l^- \nu \\ A3 \quad Z_H &\rightarrow Z H \rightarrow q\bar{q} Z Z \rightarrow q\bar{q} l^+ l^- l^+ l^- \\ A4 \quad W_H &\rightarrow W H \rightarrow qq' Z Z \rightarrow q\bar{q} l^+ l^- l^+ l^- \end{aligned} \right\} \rightarrow \text{Valence} \\
 & \left. \begin{aligned} B1 \quad Z_H &\rightarrow Z H \rightarrow l^+ l^- Z Z \rightarrow l^+ l^- q\bar{q} l^+ l^- \\ B2 \quad Z_H &\rightarrow Z H \rightarrow l^+ l^- W W \rightarrow l^+ l^- qq' l \nu \\ B3 \quad W_H &\rightarrow W H \rightarrow l \nu Z Z \rightarrow l \nu q\bar{q} l^+ l^- \end{aligned} \right\} \rightarrow \text{LAL}
 \end{aligned} \tag{3.10}$$

masse (GeV)	$m_H = 120 \text{ GeV}$	$m_H = 200 \text{ GeV}$
$B(H \rightarrow b\bar{b})$	68.3%	0.27%
$B(H \rightarrow \tau\bar{\tau})$	6.8%	0.30%
$B(H \rightarrow \gamma\gamma)$	0.22%	$\simeq 0$
$B(H \rightarrow WW)$	13%	73.5%
$B(H \rightarrow ZZ)$	1.5%	26.1%

TABLE 3.3: Rapports d'embranchement du boson de Higgs pour $m_H = 120 \text{ GeV}$ et $m_H = 200 \text{ GeV}$ [26].

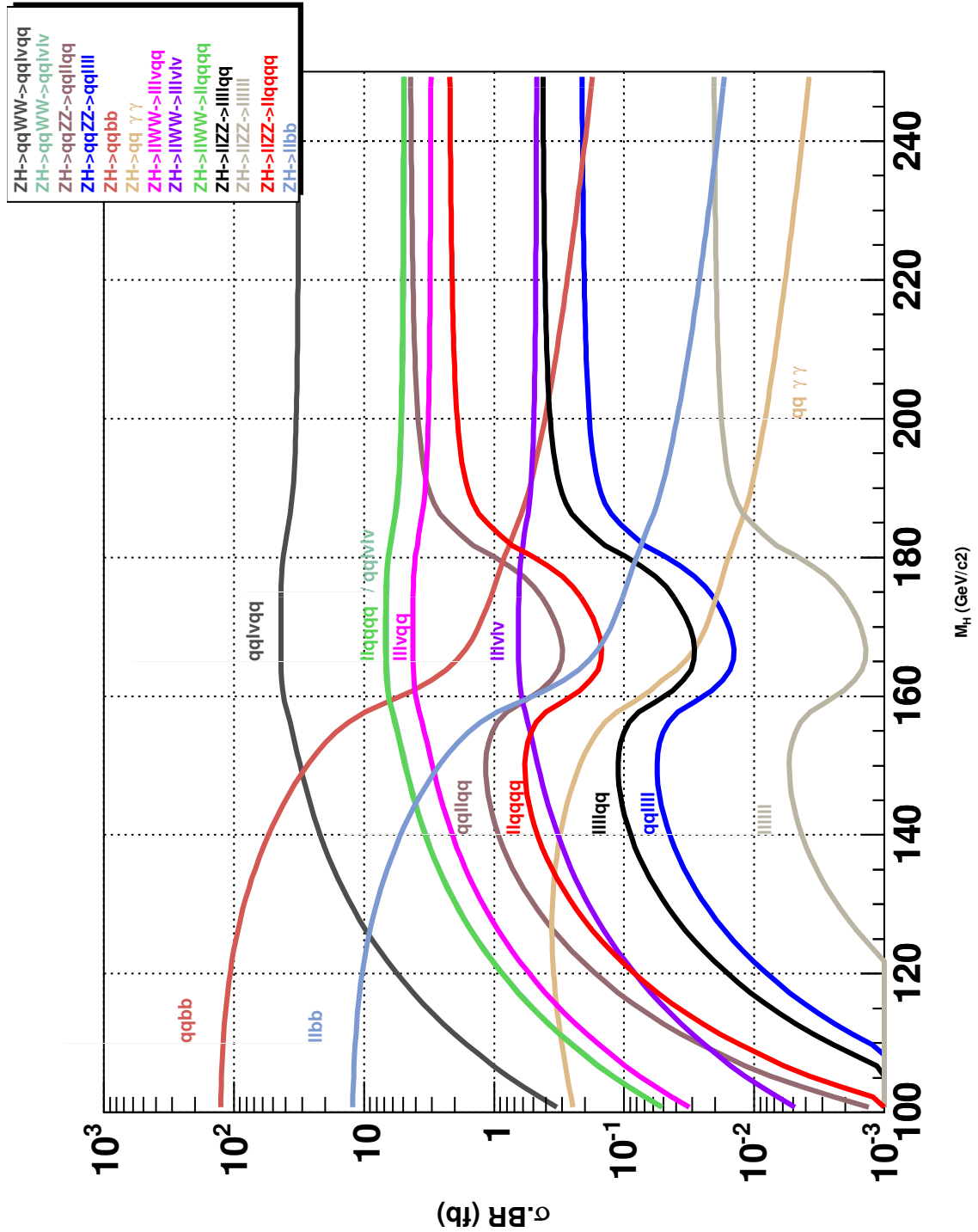


FIGURE 3.26: Section efficace multipliée par rapport d'embranchement des états finaux provenant de $Z_H \rightarrow Z H$, pour $M_{Z_H} = 1 \text{ TeV}$ et $\cot \theta = 0.5$, en fonction de la masse du boson de Higgs. [9]

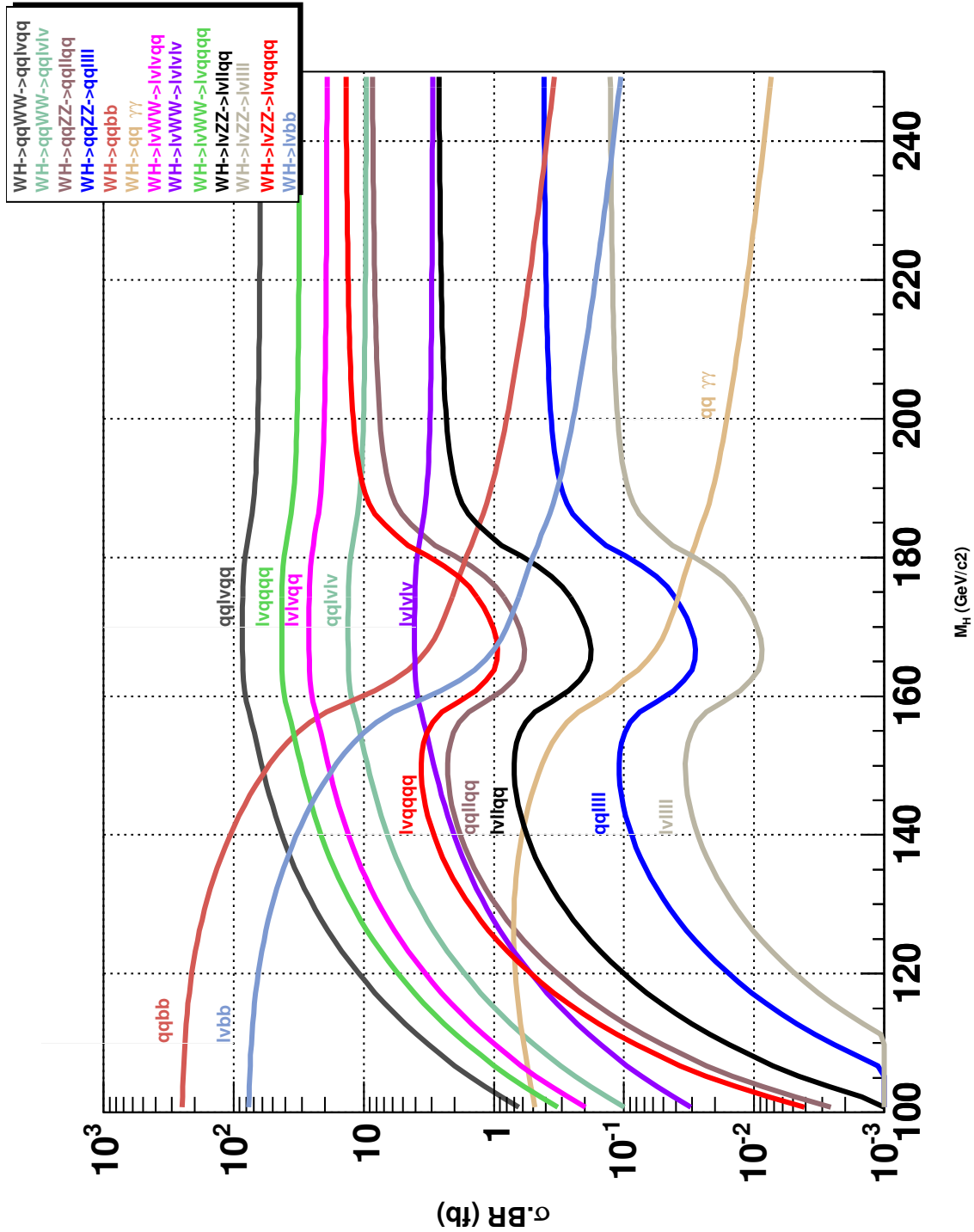


FIGURE 3.27: Section efficace multipliée par rapport d'embranchement des états finaux provenant de $W_H \rightarrow W H$, pour $M_{V_H} = 1$ TeV et $\cot \theta = 0.5$, en fonction de la masse du boson de Higgs [9]

3.2.2 Le boson de jauge A_H

Le boson de jauge A_H est plus modèle-dépendant que Z_H et W_H . La largeur partielle $\Gamma(A_H \rightarrow ZH)$ dépend de nombreux paramètres. La Figure 3.28 représente d'une part la section efficace de production de A_H en fonction de sa masse lorsque $\tan \theta' = 1$ et le cas le plus simple où $y_e = 3/5$ et $y_u = -2/5$, et d'autre part les rapports d'embranchement en fonction de $\tan \theta'$, avec $M_{A_H} = 1$ TeV et $y_e = y_u$.

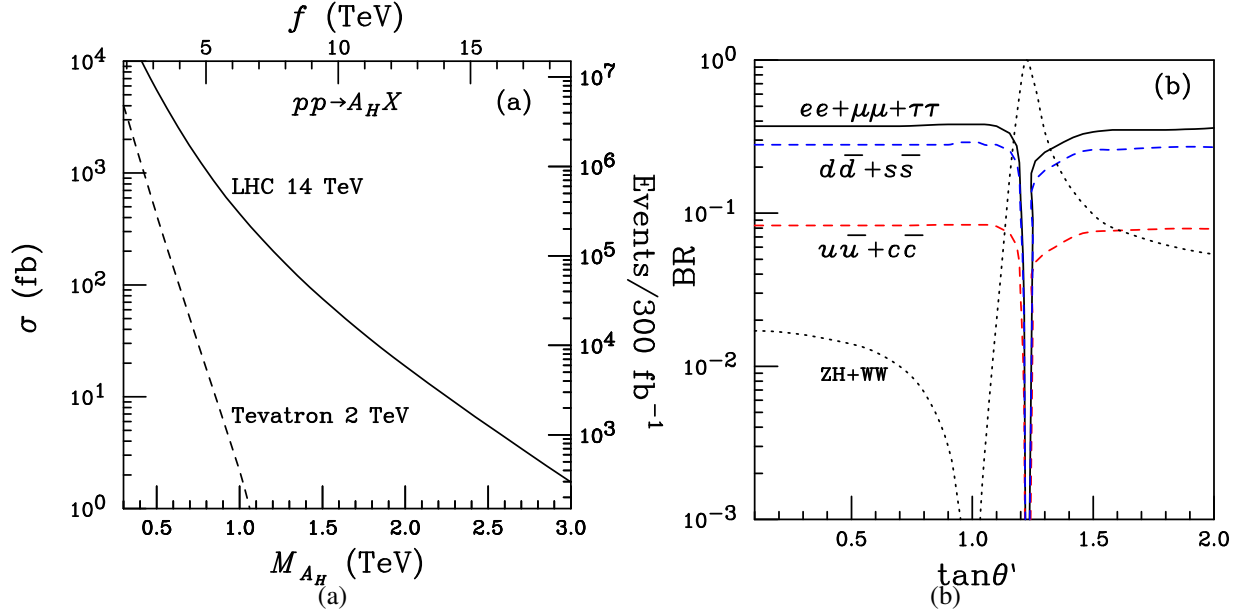


FIGURE 3.28: (a) Section efficace de production du boson de jauge lourd A_H , en fonction de sa masse et de f , pour $\tan \theta' = 1$ [16] (b) Rapports d'embranchement du boson de jauge lourd A_H en fonction de $\tan \theta'$, pour $y_e = 3/5$ et $y_u = -2/5$ [16]

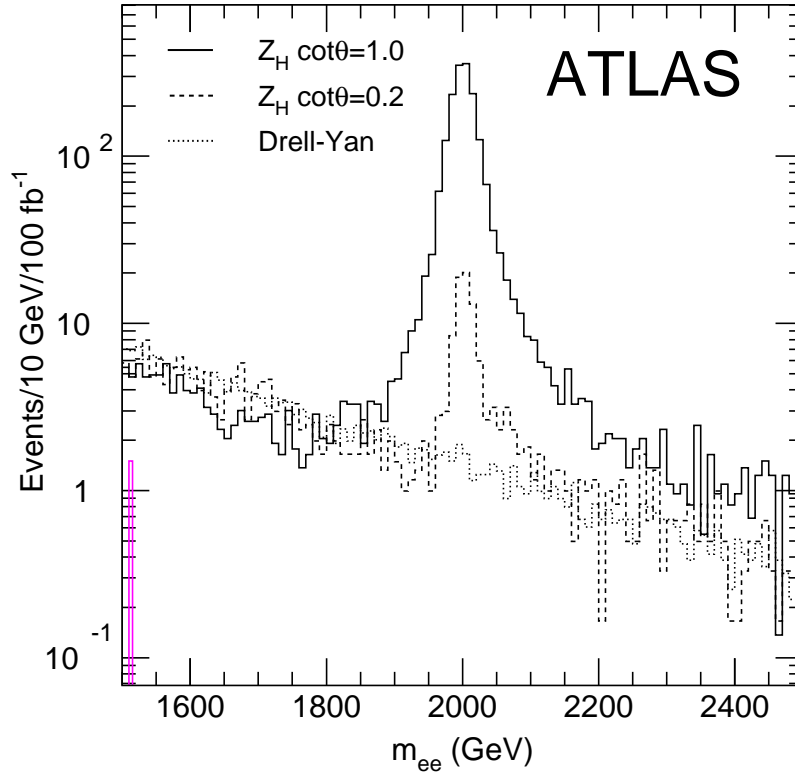


FIGURE 3.29: La distribution de masse e^+e^- pour un Z_H de masse 2 TeV pour $\cot \theta = 1$ (traits plein) et $\cot \theta = 0.2$ (tirets). L'histogramme pointillé montre la distribution du bruit de fond.

3.2.3 Découverte de Z_H , A_H et W_H

La recherche d'un pic dans la distribution de e^+e^- ou $\mu^+\mu^-$ est sensible à la présence de A_H ou Z_H . Figure 3.29 montre la distribution de masse e^+e^- venant d'un Z_H de 2 TeV pour $\cot \theta = 1$ et $\cot \theta = 0.2$. La section efficace de production pour ces deux cas est de 1.2 et 0.05 pb [16]. Les événements ont été sélectionnés en réclamant des électrons isolés de $p_T > 20$ GeV et $|\eta| < 2.5$. Le bruit de fond du modèle standard montré sur ce graphe vient du processus de Drell-Yan. Pour prétendre établir un signal, la contrainte est d'avoir au moins 10 événements avec une signification supérieure à 5σ .

Inclure le canal $\mu + \mu^-$ améliore légèrement la sensibilité, étant donné la résolution en masse moins bonne à très grande énergie. Figure 3.45 page 90 montre la région accessible en fonction de $\cot \theta$ et M_{Z_H} . Une recherche similaire de A_H peut être réalisée ; la région accessible en fonction de $\tan \theta'$ et M_{A_H} est montrée Figure 3.45. Seule une petite région autour de $\tan \theta' \sim 1.3$ où les rapports d'embranchement en $\mu^+\mu^-$ et e^+e^- sont très faibles n'est pas couverte.

La désintégration $W_H \rightarrow \ell \nu$ se manifeste par des événements avec un lepton isolé de très grand P_T et une énergie manquante importante. Les événements sont sélectionnés en demandant un électron isolé avec $p_T > 200$ GeV, $|\eta| < 2.5$ et $E_T^{miss} > 200$ GeV. La masse transverse est reconstruite dans la distribution de laquelle le signal apparaît. Le bruit de fond principal est la production de $\ell \nu$ par un W virtuel. Pour prétendre établir un signal, la contrainte est d'avoir

au moins 10 événements avec une signification supérieure à 5σ . Figure 3.45 montre la région accessible en fonction de $\cot \theta$ et M_{W_H} .

3.2.4 Observation de $Z_H \rightarrow ZH$, $A_H \rightarrow ZH$ et $W_H \rightarrow WH$ pour $m_H = 120$ GeV

L'observation des cascades $Z_H \rightarrow ZH$, $A_H \rightarrow ZH$, et $W_H \rightarrow WH$ pourrait donner une preuve cruciale que le boson de jauge lourd observé est du type prédit par les modèles du petit Higgs. Pour une masse de l'ordre de 120 GeV, deux signatures ont été recherchées : $H \rightarrow b\bar{b}$, avec un rapport d'embranchement de 68% [21], en exigeant une désintégration leptonique du W ou Z primaire, et $H \rightarrow \gamma\gamma$, avec un rapport d'embranchement beaucoup plus faible de 0.2% mais compensé par une signature plus claire [9], permettant d'utiliser les désintégrations hadroniques du W ou Z primaire. Comme indiqué plus haut, l'analyse de référence se fera pour une valeur typique de $\cot \theta \sim 0.5$, en rappelant que les rapports d'embranchement du boson de Higgs restent standard.

3.2.4.1 Bruits de fond pour $H \rightarrow \gamma\gamma$

Les bruits de fond considérés sont :

- Higgs inclusif : il s'agit de tous les modes usuels produisant au LHC un Higgs, ce dernier se désintégrant en deux photons. Ils ont été produits dans Pythia de façon standard, avec cependant la contrainte d'avoir un Higgs avec une impulsion transverse supérieure à 200 GeV de façon à obtenir une statistique suffisante d'événements similaires au signal. Les sections efficaces de ces processus avec la condition $p_T(H) > 200$ GeV sont indiquées dans la Table 3.4.

	Section efficace (fb)
$f\bar{f} \rightarrow Z^0 H$	0.077
$f_i \bar{f}_j \rightarrow W^\pm H$	0.138
$q\bar{q} \rightarrow gH$	0.010
$qg \rightarrow qH$	0.422
$gg \rightarrow gH$	0.488
$gg \rightarrow Q\bar{Q}H$	0.110
$q\bar{q} \rightarrow Q\bar{Q}H$	0.079
$f_i \bar{f}_j \rightarrow f_i \bar{f}_j H$	0.194
$f_i \bar{f}_j \rightarrow f_k \bar{f}_l H$	0.404
total	1.922

TABLE 3.4: Sections efficaces des modes produisant un Higgs, avec $p_T(H) > 200$ GeV

- $\gamma\gamma$ inclusif : il s'agit de tous les modes produisant au LHC une paire de photons, à l'exception des modes du Higgs inclusif. Pythia ne donnant pas d'assez bon résultats pour la production directe de photons à grande impulsion transverse, le générateur Monte-Carlo DIPHOX [27] lui a été préféré. La Figure 3.30 montre la distribution des paires de photons simulées en fonction de leur impulsion transverse, et indique les sections efficaces en fonction du seuil sur $p_T(\gamma\gamma)$. Les photons ont été générés avec les conditions suivantes :
 - $|\eta_\gamma| < 2.5$
 - 1 photon avec $p_T > 25$ GeV, l'autre avec $p_T > 40$ GeV
 - $|m(\gamma\gamma) - 120| < 5$ (en GeV)

Les deux premières sont des coupures standard de détection (voir plus loin), et la dernière permet de ne garder que les événements susceptibles de se rapprocher du signal. DIPHOX n'a pas été interfacé avec Pythia si bien qu'il n'est pas possible de simuler des événements complets.

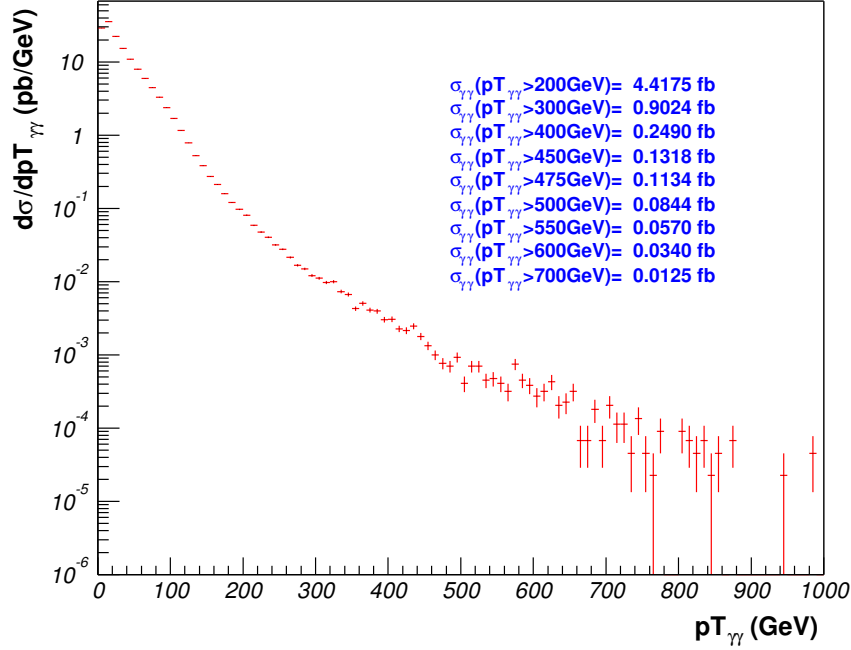


FIGURE 3.30: Section efficace de production de paires $\gamma\gamma$ en fonction de l'impulsion transverse, par unité d'impulsion transverse de $\gamma\gamma$ prédite par DIPHOX (voir texte pour les coupures)

- $\gamma - jet$: il s'agit du bruit de fond combinant un photon authentique avec un jet mal-identifié. Ce bruit de fond est connu pour être à basse impulsion au moins deux fois plus petit que le $\gamma\gamma$ inclusif et pour diminuer à grande impulsion plus vite que celui-ci. Il sera donc négligé.

3.2.4.2 Canal $V_H \rightarrow V(q\bar{q})H(\gamma\gamma)$

La désintégration $H \rightarrow \gamma\gamma$ donne un signal caractéristique, puisqu'elle suffit à donner un canal de découverte du boson Higgs dans le modèle standard. Une sélection préliminaire est de requérir deux photons isolés avec $|\eta| < 2.5$, avec $p_T > 25$ GeV et au moins un photon avec $p_T > 40$ GeV. Cette contrainte est suffisante pour le déclenchement. La masse invariante des deux photons doit être à moins de 2σ de la masse du boson de Higgs (qui aura été découvert auparavant).

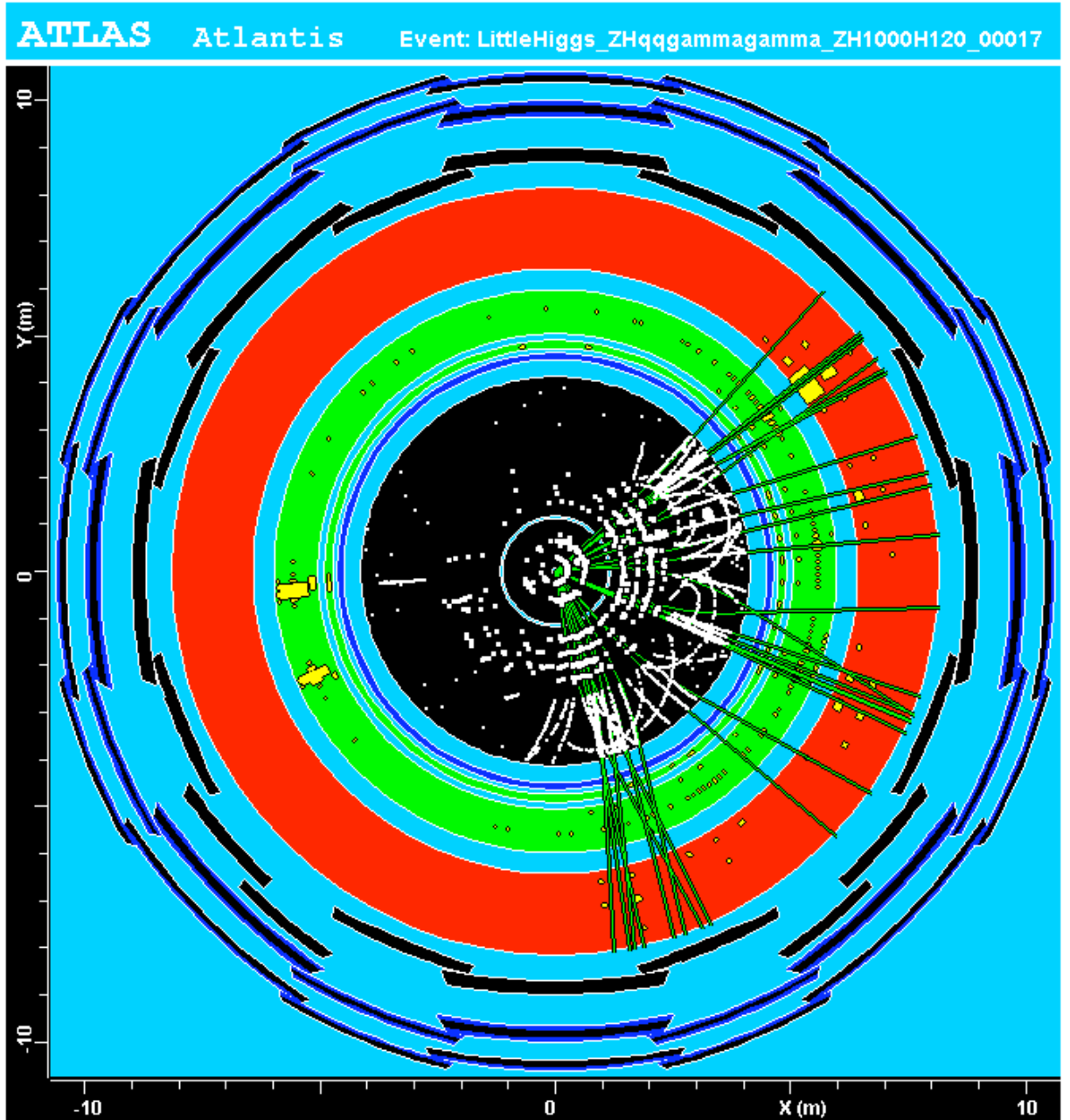


FIGURE 3.31: Représentation d'un événement du signal $Z_H \rightarrow q\bar{q}\gamma\gamma$ avec $M_{Z_H} = 1$ TeV. Sont visibles les 2 photons d'un côté (à gauche, sans trace associée) et les jets de l'autre

La difficulté de ces canaux vient de la reconstruction des désintégrations hadroniques $Z \rightarrow q\bar{q}$ et $W \rightarrow q\bar{q}$ alors que l'impulsion transverse typique des W et Z est de 500 GeV

pour $m_{V_H} = 1$ TeV, ou 1 TeV pour $m_{V_H} = 2$ TeV. Figure 3.31 affiche une représentation d'événements dans un ca où les jets sont bien séparés. Mais à ces très hautes impulsions les deux jets résultant des quarks primaires peuvent être mal séparés, nécessitant la mise au point d'un algorithme particulier :

- parmi toutes les paires de jets qu'il est possible de composer et ayant une masse invariante autour de M_Z , plus précisément telle que $|m(jj) - M_Z| < 30$ GeV, seule celle ayant la plus grande impulsion transverse est sélectionnée. Il faut noter que cette condition concerne les deux signaux Z_H et W_H , car la résolution en masse di-jet ne permet pas de les séparer. Le choix s'est porté sur m_Z pour s'éloigner du bruit de fond à basse masse.
- si l'impulsion transverse de cette paire est supérieure $p_{T\ min} = 200$ GeV, alors cette paire est considérée comme étant celle qui reconstruit le Z ou W , et l'énergie des 2 jets est alors corrigée de telle sorte que leur masse invariante soit exactement égale à M_Z
- si au contraire l'impulsion transverse de la paire est inférieure à $p_{T\ min}$ il est probable que la paire ne corresponde pas à un Z ou W et que le boson vecteur soit plutôt détecté en un seul jet, de plus grande énergie. Ce jet unique est alors choisi comme étant le jet avec la plus grande impulsion transverse. Ce cas de figure représente environ 20 % des cas pour $M_{V_H} = 1000$ GeV et environ 60% pour $M_{V_H} = 2000$ GeV. l'impulsion transverse du Z et du W ainsi reconstruits est représentée sur la Figure 3.32.

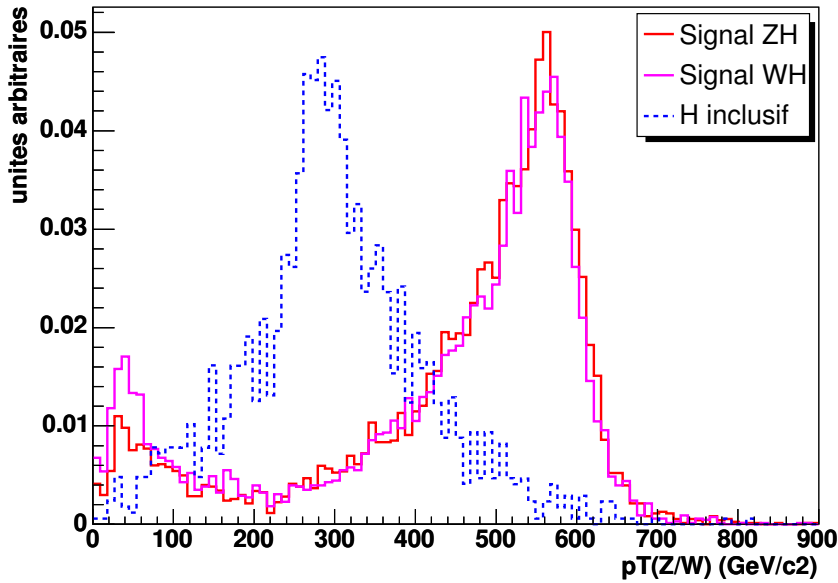


FIGURE 3.32: Impulsion transverse de la paire de jets ou du jet unique reconstruisant le Z ou W , pour les deux signaux et le bruit de fond Higgs inclusif, pour $M_{V_H} = 1000$ GeV (distribution normalisée obtenue après les coupures cinématiques standard et la coupure sur la masse reconstruite du Higgs)

- la masse de Z_H et W_H est finalement reconstruite à partir de la paire de jets sélectionnée ou bien du jet unique, combiné(e) à la paire de photons

Cette méthode est celle qui conduit à une résolution optimale de la masse invariante de Z_H et W_H . La Figure 3.33 compare les masses reconstruites pour les trois cas suivants :

- reconstruction avec 2 jets

- reconstruction avec 1 seul jet
- reconstruction mixte décrite ci-dessus avec comme paramètre de choix $p_{T \min}$

La reconstruction avec 2 jets montre un excès pour une masse reconstruite nettement inférieure au pic principal, correspondant à une partie des événements où cette reconstruction avec 2 jets est fortuite. La reconstruction avec 1 seul jet montre une résolution dégradée par rapport au pic du premier cas, l'épaule à gauche du pic correspondant à la perte d'un jet dans la reconstruction finale. La reconstruction mixte apparaît ainsi comme un bon compromis, même si le bruit de fond du boson de Higgs inclusif est plus élevé.

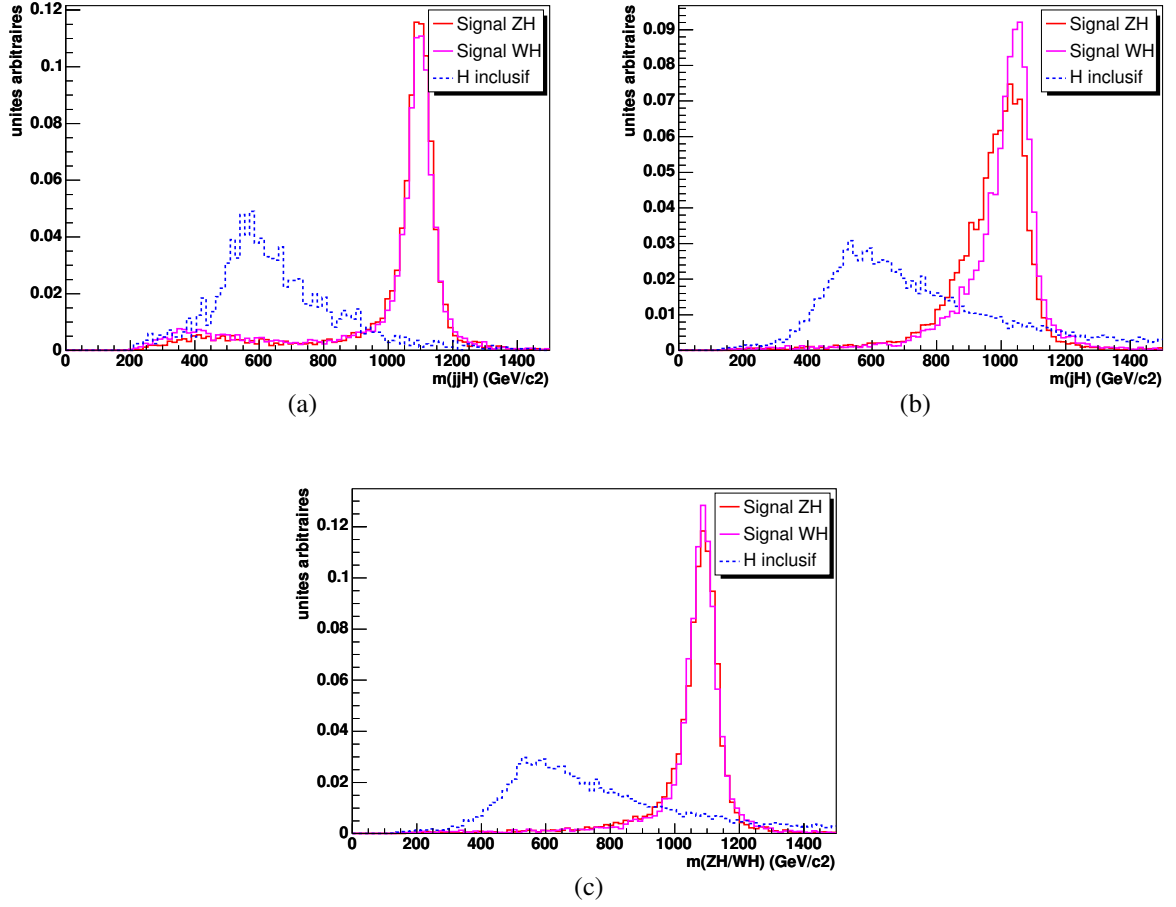


FIGURE 3.33: Masse reconstruite de V_H du signal et du bruit de fond Higgs inclusif, pour $M_{V_H} = 1000$ GeV (distribution normalisée) dans le cas d'une reconstruction avec 2 jets (a), un seul jet (b), ou une méthode mixte (c) finalement retenue

La résolution sur la masse du V_H est de 50 GeV pour $m_{V_H} = 1$ TeV, 80 GeV pour $m_{V_H} = 2$ TeV. La masse reconstruite est montrée sur Figure 3.34. L'efficacité totale de reconstruction est de 32% et dépend peu de M_{V_H} . La contribution des W_H et Z_H sont montrées séparément mais elles ne peuvent pas être séparées expérimentalement la résolution en masse étant supérieure à la différence de masse des W et Z , soit 10 GeV. La contribution du W_H domine étant donné sa section efficace double.

La présence d'une paire de photons formant une masse compatible avec celle du boson de Higgs permet un bruit de fond réduit. Le bruit de fond vient alors soit de la production standard

de boson de Higgs, soit de la production QCD de di-photons. Ce dernier bruit de fond a été simulé avec le Monte-Carlo dédié DIPHOX. En l'absence d'événements complets simulés avec DIPHOX, l'efficacité des coupures mettant en jeu les jets est supposée être identique au cas du Higgs inclusif. Une dernière coupure est appliquée sur l'impulsion transverse de la paire de photons. Cette coupure est optimisée de 300 GeV pour $m_{V_H} = 850$ GeV à 700 GeV pour $m_{V_H} = 2$ TeV.

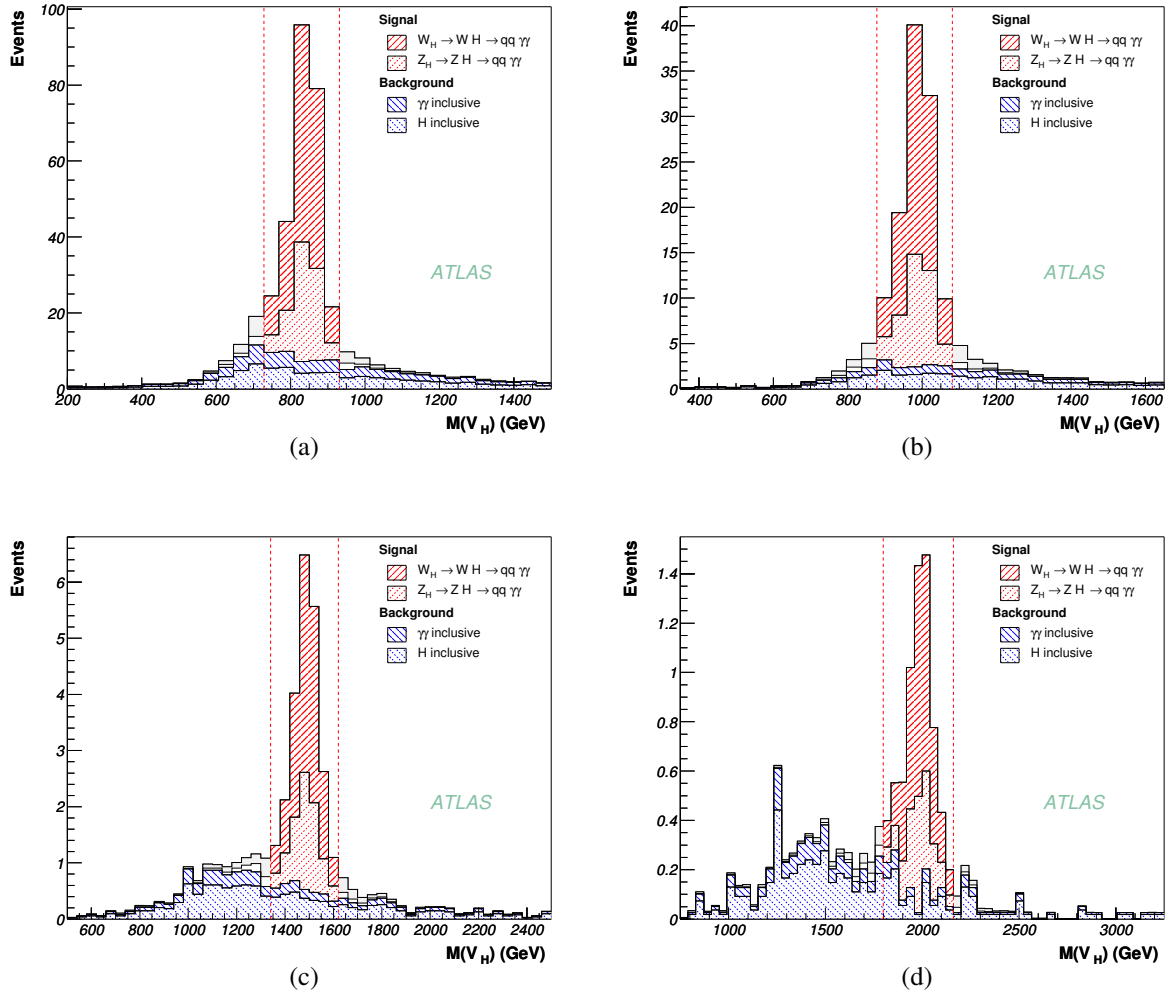


FIGURE 3.34: Masse reconstruite de V_H du signal et du bruit de fond, pour $M_{V_H} = 850$ (a), 1000 (b), 1500 (c) et 2000 (d) GeV, à 300 fb^{-1}

Le Tableau 3.5 présente les niveaux de signal et de bruit ainsi que la signification statistique, après toutes les coupures pour les différentes masses, en supposant une luminosité intégrée de 300 fb^{-1} (soit trois ans de fonctionnement du LHC à haute luminosité). Le niveau de bruit de fond est toujours inférieur à 20 % du signal. Etant donnés les petits nombres d'événements, la signification statistique n'est pas simplement S/\sqrt{B} mais est calculée en utilisant la loi de Poisson.

La signature $H \rightarrow \gamma$ est si claire qu'il n'est pas absolument nécessaire de reconstruire complètement la désintégration $V_H \rightarrow VH$. Cela permet de gagner en signal puisqu'il n'y a pas

M_{V_H} (GeV)		850	1000	1250	1500	1750	2000
$p_T(\gamma\gamma)_{min}$ (GeV)		300	400	500	500	550	700
Signal	W_H	142.2	64.2	28.8	12.7	6.8	3.6
	Z_H	73.3	33.0	14.5	6.3	3.5	1.8
	total S	215.5 ± 2.2	97.2 ± 1.0	43.3 ± 0.5	19.1 ± 0.2	10.3 ± 0.1	5.4 ± 0.1
Bruit	H	20.2	8.2	3.1	2.3	1.3	0.7
	$\gamma\gamma$	15.1	4.5	1.4	1.0	0.5	0.3
	total B	35.3 ± 0.6	12.7 ± 0.4	4.4 ± 0.3	3.3 ± 0.2	1.2 ± 0.1	1.0 ± 0.1
Signification statistique		23.4 ± 0.1	16.6 ± 0.1	$11.7^{+0.1}_{-0.2}$	6.6 ± 0.1	5.1 ± 0.1	3.0 ± 0.1

TABLE 3.5: Nombre d'événements de signal et de bruit de fond, et signification statistique, pour $V_H \rightarrow V(qq) + H(\gamma\gamma)$, à 300 fb^{-1}

de perte en efficacité ni en rapport d'embranchement dans la reconstruction du V.

L'analyse et les coupures précédentes restent valables (mise à part évidemment la coupure sur la masse reconstruite de V_H), et le signal est visible (voir Figure 3.35). Cette méthode, même si elle conduit à un niveau de bruit de fond plus élevé, donne une signification statistique légèrement meilleure, comme le prouve le Tableau 3.6. Toutefois, malgré la similitude de la signification statistique, la reconstruction d'un pic de masse de V_H sera plus convaincante et plus robuste qu'un excès de Higgs à grande impulsion transverse. Ce résultat ne sera donc pas utilisé dans la suite.

M_{V_H} (GeV)		850	1000	1500	2000
$p_T(\gamma\gamma)_{min}$ (GeV)		300	400	550	700
Signal	W_H	236.4	104.6	19.8	5.8
	Z_H	120.4	53.5	9.9	2.9
	total S	356.8 ± 3.3	158.1 ± 1.6	29.7 ± 0.3	8.7 ± 0.1
Bruit	H	81.8	30.6	8.2	2.7
	$\gamma\gamma$	61.3	16.7	3.6	0.7
	total B	143.1 ± 1.3	47.2 ± 0.8	11.8 ± 0.4	3.4 ± 0.2
Signification statistique		29.8 ± 0.3	23.0 ± 0.3	8.7 ± 0.2	4.7 ± 0.2

TABLE 3.6: Nombre d'événements de signal et de bruit de fond, et signification statistique, pour une reconstruction inclusive de $V_H \rightarrow V(X) + H(\gamma\gamma)$, à 300 fb^{-1} . Ces résultats ne seront pas utilisés dans la suite.

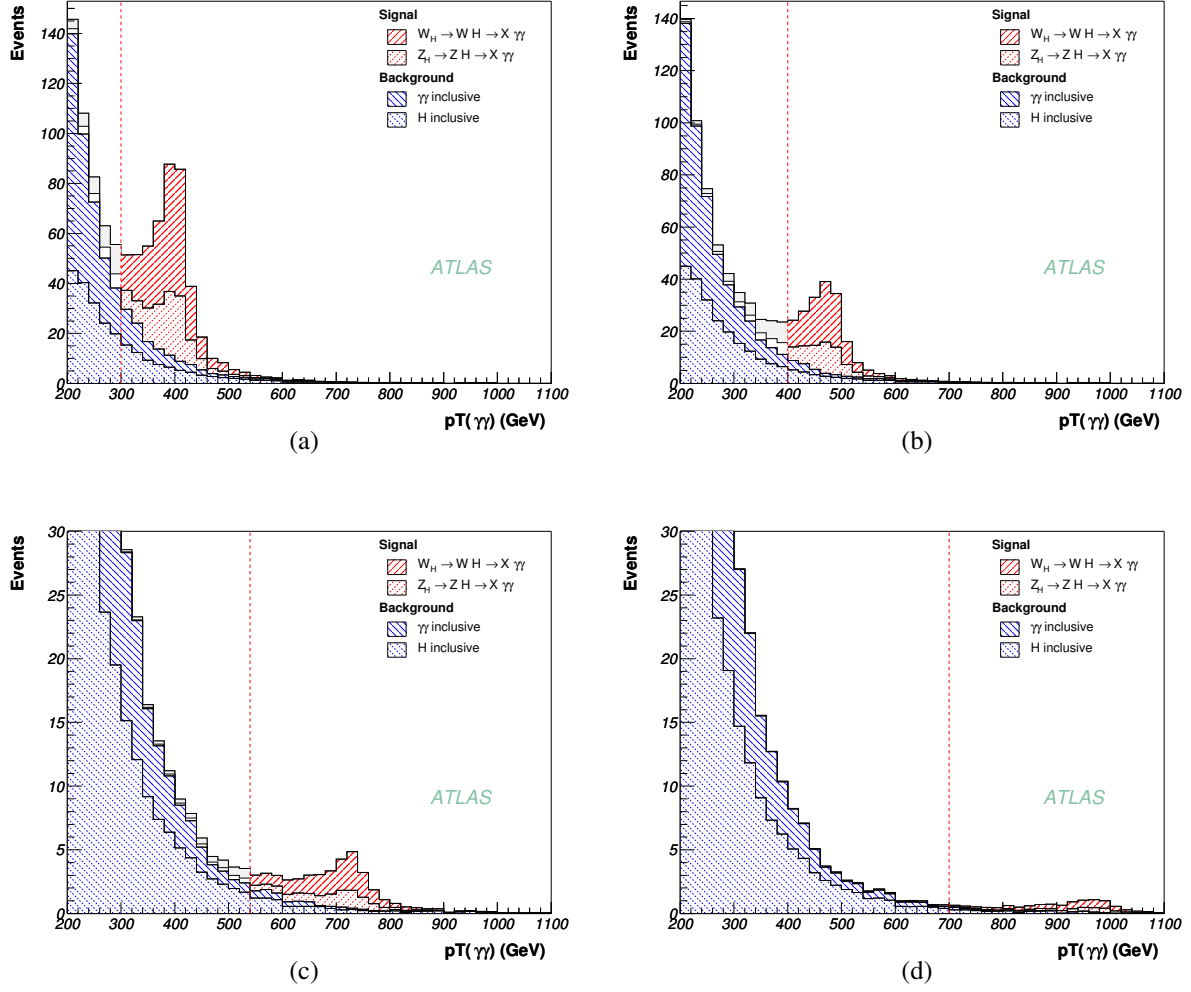


FIGURE 3.35: Impulsion transverse du Higgs pour le signal et le bruit de fond, pour $M_{V_H} = 850$ (a), 1000 (b), 1500 (c) et 2000 (d) GeV, à 300 fb^{-1}

3.2.4.3 Canal $V_H \rightarrow VH(b\bar{b})$

M_{V_H} (GeV)		1000	2000
Signal 1	Z_H	195	15.0
	bruit $Z + jets$	15.5	8.3
	Signification statistique	26.5	4.0
Signal 2	W_H	1499	179
	bruit $W + jets$	250	203
	bruit $t\bar{t}$	221	309
	Signification statistique	39	7.5

TABLE 3.7: Nombre d'événements de signal et de bruit de fond, et signification statistique, pour $V_H \rightarrow V(leptons) + H(b\bar{b})$, à $300 fb^{-1}$

Les états finaux considérés sont $Z_H \rightarrow Z(\ell^+\ell^-)H(b\bar{b})$ et $W_H \rightarrow W(\ell\nu)H(b\bar{b})$. La désintégration en cascade $ZH \rightarrow \ell^+\ell^-b\bar{b}$ donne un état final avec deux b -jets reconstruisant la masse du boson de Higgs, et une paire $\ell^+\ell^-$ reconstruisant la masse du Z , et donnant le déclenchement. Le boson de Higgs ayant une impulsion transverse de l'ordre de la moitié de la masse du V_H , le problème de la séparation des jets correspondants aux quarks b se pose comme dans le cas précédent. Etant donné que l'identification des jets de b est importante pour réduire le bruit de fond, deux algorithmes distincts ont été utilisés : identification séparée des jets de b à 1 TeV, et identification d'un seul "gros" jet issus des deux quarks b à 2 TeV.

La sélection des événements $Z_H \rightarrow ZH(b\bar{b})$ est faite comme suit :

- deux leptons de charge opposées et de même saveur avec $p_T > 6$ (resp. 5 GeV) pour les muons (resp. électrons) et $|\eta| < 2.5$. L'un d'eux doit avoir $p_T > 25$ GeV et fournit le déclenchement.
- la paire de lepton doit avoir une masse reconstruite entre 76 et 106 GeV.
- Pour $m_{Z_H} = 1$ TeV :
 - deux jets étiquetés b avec $p_T > 25$ GeV et $|\eta| < 2.5$, et séparés de moins de $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} < 1.5$.
 - la paire de b -jets doit avoir une masse reconstruite entre 60 et 180 GeV
- Pour $m_{Z_H} = 2$ TeV :
 - un seul jets étiqueté b avec $p_T > 500$ GeV et $|\eta| < 2.5$

L'efficacité pour $M_{Z_H} = 1$ TeV est de 6 % augmentant à 10 % pour une masse de 2 TeV, car la sélection d'un seul gros b -jet est plus efficace que la sélection de deux b -jets, mais rejette moins de bruit de fond. La masse reconstruite du système ZH est montrée sur la Figure 3.36. Le bruit de fond vient principalement d'états finaux $Z + jet$.

Une méthode similaire permet de reconstruire la désintégration $W_H \rightarrow WH \rightarrow \ell\nu b\bar{b}$. La sélection de la paire de b -jet ou d'un unique gros b -jet reste identique, tandis que maintenant la sélection des leptons vérifie :

- un e ou μ isolé avec $p_T > 25$ GeV et $|\eta| < 2.5$.
- $E_T^{miss} > 25$ GeV.

L'énergie transverse est supposée ne venir que du neutrino de la désintégration leptonique du W , ce qui permet de reconstruire l'impulsion du W . L'efficacité pour $M_{W_H} = 1$ TeV est 7 %

augmentant à 12 % pour 2 TeV. Le bruit de fond, dominé par les événements $W + jets$ et $t\bar{t}$ est plus important que dans le cas précédent mais le signal est néanmoins visible (voir Figure 3.36. Le nombre d'événements reconstruits, signal, bruit de fond et signification statistique dans les deux cas est indiqués Tableau 3.7.

Contrairement au canal $H \rightarrow \gamma\gamma$, le canal $H \rightarrow b\bar{b}$, ne se prête pas à une reconstruction inclusive, l'identification des b-jets n'étant pas assez bonne. En effet le taux de misidentification d'un photon est de l'ordre de 1/10000 pour une efficacité de 80%, tandis que pour un b-jet de grande impulsion elle est de l'ordre de 1/100 (voire 1/10 pour un jet charmé) pour une efficacité de 50%.

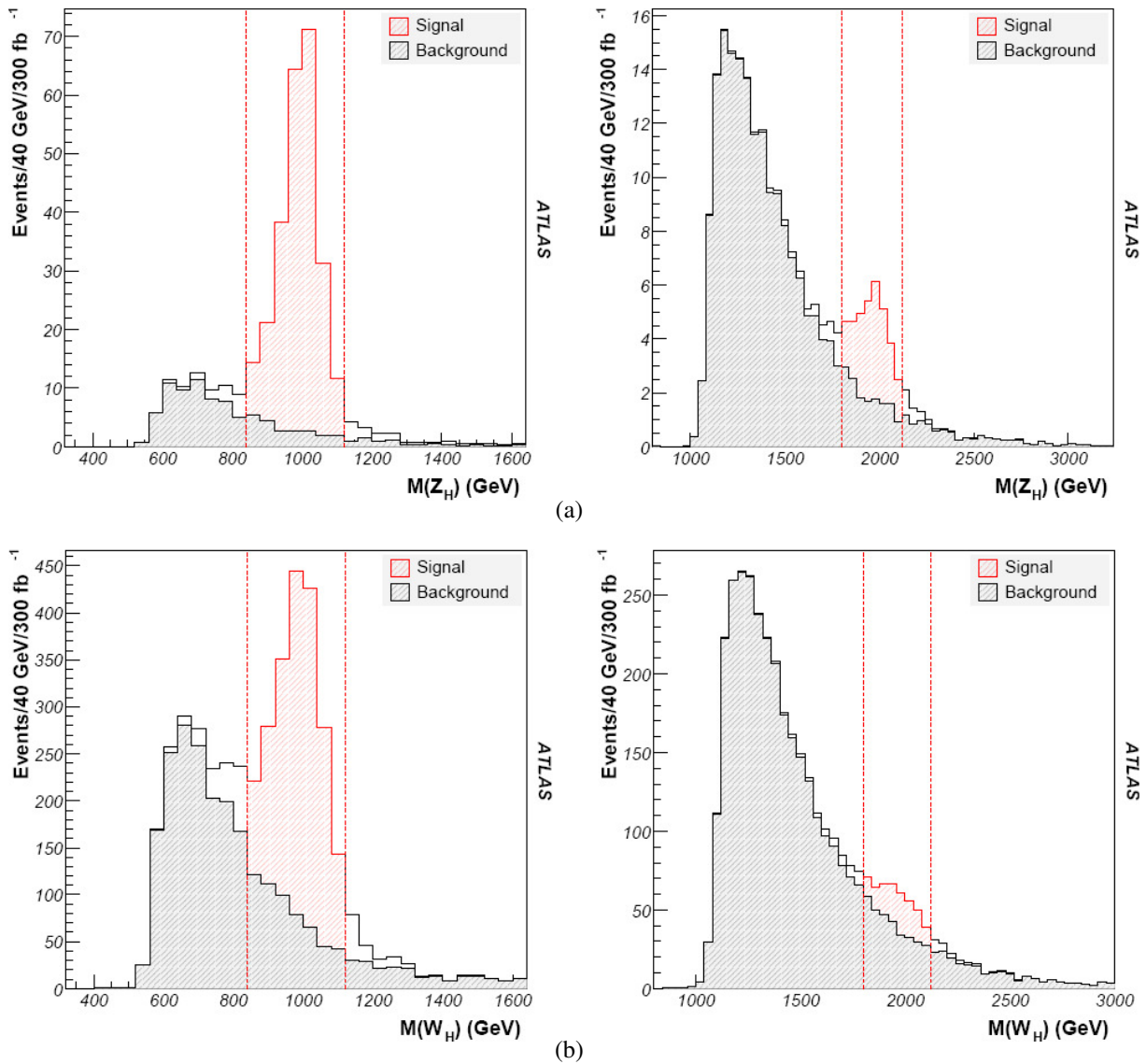


FIGURE 3.36: Masse reconstruite pour le signal et le bruit de fond, pour l'état final $Z_H \rightarrow Z(l^+l^-)H(b\bar{b})$ (a) avec $m_{Z_H} = 1000$ (gauche) et 2000 (droite) GeV, et pour l'état final $W_H \rightarrow W(l\nu)H(b\bar{b})$ (b) avec $m_{W_H} = 1000$ (gauche) et 2000 (droite) GeV, à 300 fb^{-1}

3.2.4.4 Observabilité et généralisation

Les régions accessibles sont montrées Figure 3.45. Si $m_H = 120$ GeV, la masse des bosons lourds doit être inférieure à 2 TeV comme indiqué plus haut. Une fraction assez importante de l'espace des paramètres est couverte, sauf pour la région autour de $\cot \theta = 1$.

La recherche de $A_H \rightarrow ZH$ est identique à la recherche $Z_H \rightarrow ZH$. Toutefois la section efficace de production de A_H et la désintégration en Z_H dépend de plusieurs paramètres si bien que l'observabilité d A_H ne peut pas être représentée aisément en termes de ces paramètres. La section 3.2.7 donne les limites indépendantes des modèles en terme de section efficace multiplié par les rapports d'embranchement.

3.2.5 Observation de $Z_H \rightarrow ZH$ et $W_H \rightarrow WH$ pour $m_H = 200$ GeV

Si le boson de Higgs a une masse de 200 GeV les principaux canaux de désintégrations sont $H \rightarrow W^+W^-$ (73 %) et $H \rightarrow ZZ$ (26 %). Les différents états finaux ZH et WH sélectionnés sur la base d'un compromis entre la section efficace et la clarté de la signature, sont listés dans la Table 3.8. Pour les modes A, étudiés à l'université de Valence [21], tous les leptons sont isolés et l'état final du boson de Higgs est purement leptonique. Pour les modes B, étudiés au LAL [9], l'état final du boson de Higgs contient une désintégration hadronique des W ou Z , qui est susceptible de masquer les autres leptons.

TABLE 3.8: Les états finaux W_H and Z_H étudiés pour $m_H = 200$ GeV. Les «leptons» incluent électrons et muons. Les rapports d'embranchement sont calculés en supposant $\cot \theta = 0.5$.

Mode	BR (10^{-4})	désintégration	signature
A1 :	0.97	$Z_H \rightarrow ZH \rightarrow \ell^+\ell^-W^+W^- \rightarrow \ell^+\ell^- \ell^+\nu\ell^-\nu$	4 leptons + E_T^{miss}
A2 :	3.0	$W_H \rightarrow WH \rightarrow \ell\nu W^+W^- \rightarrow \ell\nu \ell^+\nu\ell^-\nu$	3 leptons + E_T^{miss}
A3 :	0.36	$Z_H \rightarrow ZH \rightarrow qqZZ \rightarrow qq \ell^+\ell^-\ell^+\ell^-$	4 leptons + jets
A4 :	0.36	$W_H \rightarrow WH \rightarrow qqZZ \rightarrow qq \ell^+\ell^-\ell^+\ell^-$	4 leptons + jets
B1 :	0.73	$Z_H \rightarrow ZH \rightarrow \ell^+\ell^-ZZ \rightarrow \ell^+\ell^- qq\ell^+\ell^-$	4 leptons + jets
B2 :	6.2	$Z_H \rightarrow ZH \rightarrow \ell^+\ell^-W^+W^- \rightarrow \ell^+\ell^- qq\ell\nu$	3 leptons + jets + E_T^{miss}
B3 :	2.3	$W_H \rightarrow WH \rightarrow \ell\nu ZZ \rightarrow \ell\nu qq\ell^+\ell^-$	3 leptons + jets

3.2.5.1 Bruits de fond

Il existe un certain nombre de bruits de fond potentiels pour les signaux envisagés, c'est-à-dire avec le même nombre de leptons, donnés avec leur section efficace dans le Tableau 3.9 ; est aussi indiquée la statistique avec laquelle ils ont été produits dans Pythia. Il y a d'une part les canaux qui comprennent un boson de Higgs inclusif se désintégrant de la même manière que le Higgs des signaux, et qui peut être associé à un Z ou W ou des quarks, et d'autre part les productions $t\bar{t}$, ZZ et WZ .

Les bruits de fond les plus importants sont :

- $t\bar{t}$: même si la topologie n'est pas exactement identique à celle des signaux, la section efficace est tellement grande qu'il ne peut être écarté. Il a été produit en très grande quantité (2 millions d'événements Monte-Carlo) afin d'estimer le plus précisément possible son niveau après coupures.
- WZ : là encore la section efficace est très grande comparée aux signaux, et l'état final en est assez proche ; pour cette raison une grande statistique a aussi été nécessaire.

	$\sigma \cdot \text{BR (fb)}$	Statistique
$t\bar{t} \rightarrow W^+bW^-\bar{b} \rightarrow l^+\nu b l^-\nu\bar{b}$ avec b se désintégrant en lepton	3376.000	2000000
$t\bar{t}H \rightarrow W^+bW^-\bar{b} ZZ \rightarrow l^+\nu b l^-\nu\bar{b} l^+l^- q\bar{q}$	0.190	10000
$qqH \rightarrow qqZZ \rightarrow qq l^+l^- l^+l^-$	2.813	10000
$H \rightarrow ZZ \rightarrow l^+l^- l^+l^-$	46.790	100000
$fH \rightarrow fZZ \rightarrow f l^+l^- l^+l^-$	5.723	10000
$ZZ \rightarrow l^+l^- l^+l^-$	70.530	100000
$ZH(1) \rightarrow l^+l^- ZZ \rightarrow l^+l^- q\bar{q} l^+l^-$	0.228	10000
$ZH(2) \rightarrow l^+l^- WW \rightarrow l^+l^- qq' l\nu$	2.028	10000
$WH \rightarrow qq' ZZ \rightarrow qq' l^+l^- l^+l^-$	1.295	10000
$WZ \rightarrow l\nu l^+l^-$	387.600	1000000

TABLE 3.9: Section efficace et statistique de production Monte-Carlo des bruits de fond

- ZH et WH : ces bruits de fond ont le même état final que le signal.

Pour éviter que les jets masquent les leptons venus du même boson de Higgs, les critères d'isolation des leptons ont du être réduit de $\Delta R > 0.2$ standard à $\Delta R > 0.1$, pour éviter que les leptons ne soient masqués par les jets provenant du même boson de Higgs. Figure 3.37 affiche une représentation d'événement où jets et leptons issus du boson de Higgs se superpose.

3.2.5.2 Signal B1

Il s'agit du canal $Z_H \rightarrow Z_1 H \rightarrow \ell^+ \ell^- Z_2 Z_3 \rightarrow \ell^+ \ell^- \ell^+ \ell^- jj$. La reconstruction de Z_H suit le schéma indiqué ici :

- la première condition, essentielle, est de ne sélectionner que les événements ayant exactement 4 leptons
- avec ces 4 leptons sont formées 2 paires de leptons de même saveur et de charge opposée, chacune ayant une masse invariante $m(ll)$ telle que $|m(ll) - M_Z| < 5 \text{ GeV}$, de manière à reconstruire les 2 Z supposés présents dans l'événement
- parmi ces 2 paires de leptons, celle avec la plus grande impulsion transverse est choisie comme étant celle correspondant au Z primaire qui vient de Z_H , Z_1 , et l'autre reconstruit naturellement un des 2 Z provenant du Higgs, Z_2
- il reste finalement à reconstruire le dernier Z , celui issu du Higgs et se désintégrant en quarks, Z_3 :
 - parmi tous les jets reconstruits, sont sélectionnés d'une part le jet qui a la plus grande impulsion transverse, et d'autre part les paires de jets (parmi toutes les combinaisons possibles) dont la masse invariante $m(jj)$ remplit la condition $|m(jj) - M_Z| < 15 \text{ GeV}$
 - plusieurs Higgs candidats sont reconstruits en associant à Z_2 chaque paire de jets sélectionnée ainsi que le jet à impulsion transverse maximum
 - la bonne solution est déterminée par le Higgs candidat qui a la plus grande impulsion transverse parmi ceux dont la masse invariante $m(H)$ vérifie $|m(H) - M_H| < 30 \text{ GeV}$; M_H est la masse du Higgs, supposée connue
- pour finir le Z primaire (Z_1) et le Higgs reconstruits permettent d'obtenir la masse reconstruite de Z_H

Les coupures suivantes sont ensuite appliquées :

- la paire de jets ou le jet unique doit avoir une impulsion transverse plus grande que 50 GeV ; cette coupure épargne le signal tandis qu'elle écarte certains bruits de fond, comme l'illustre la Figure 3.38
- la masse reconstruite doit être à moins de 2σ de la masse du Higgs
- le Higgs doit avoir une impulsion transverse plus grande que 100 GeV ; cette coupure conduit à une suppression importante du bruit de fond quel que soit leur type, presque sans affecter le signal, comme le montre la Figure 3.39
- une coupure à 2σ sur la masse reconstruite de Z_H

Le bruit de fond est efficacement supprimé, si bien qu'il est nécessaire de l'extrapoler depuis les basses masses jusque sous le pic du signal avec une forme exponentielle. La masse reconstruite pour différentes masses de Z_H apparaissent Fig 3.40.

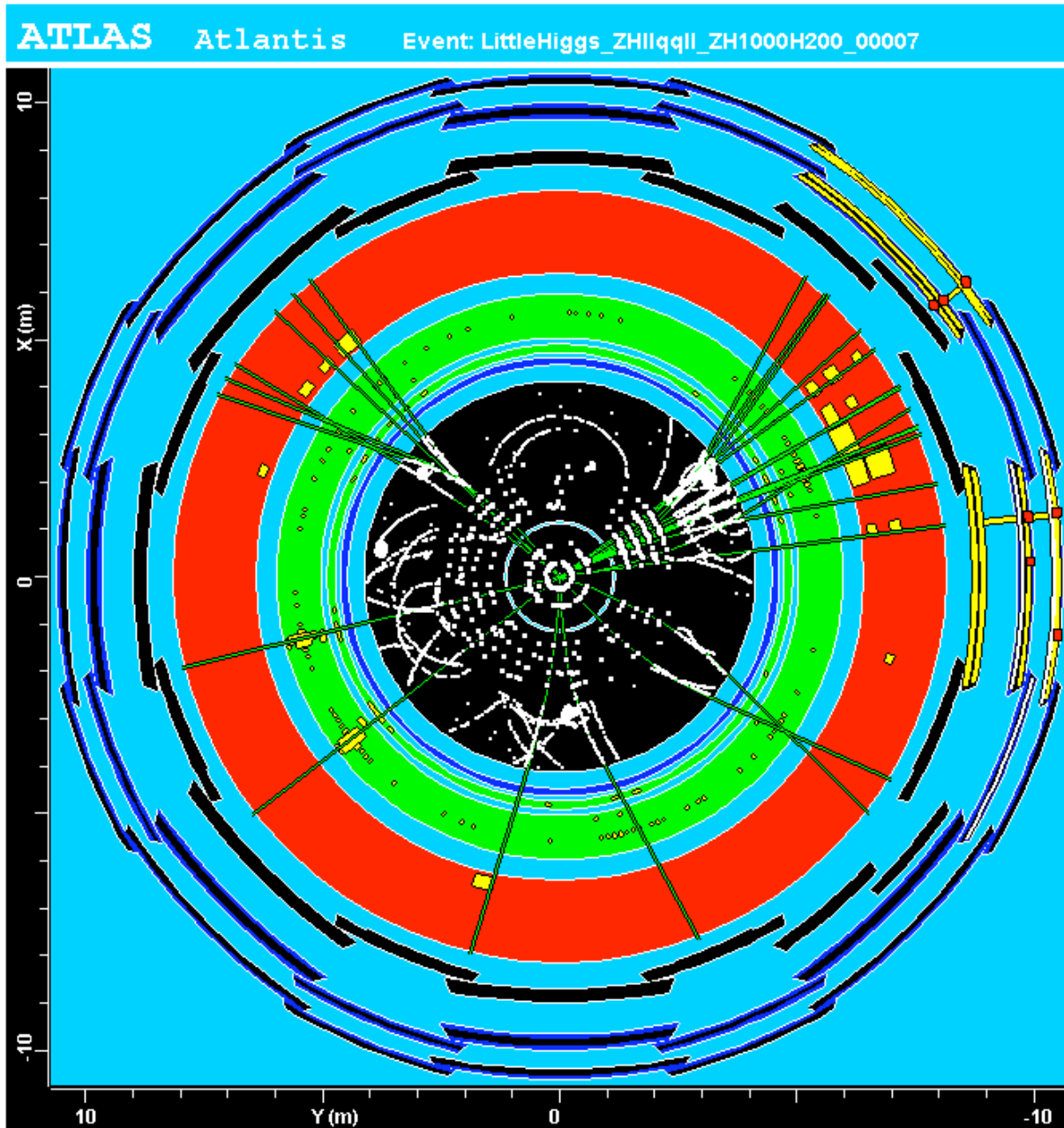


FIGURE 3.37: Représentation d'un événement du signal $Z_H \rightarrow ll \, q\bar{q}ll$ avec $M_{Z_H} = 1 \text{ TeV}$. En haut à droite sont visibles une paire de muons et les jets, issus du Higgs, et de l'autre côté une paire d'électrons

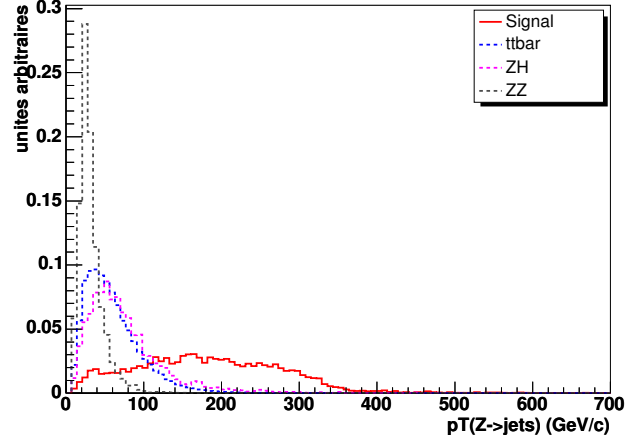


FIGURE 3.38: Impulsion transverse de la paire de jets ou du jet unique, pour le canal B1 et quelques-uns de ses bruits de fond caractéristiques, pour $M_{Z_H} = 1000$ GeV.

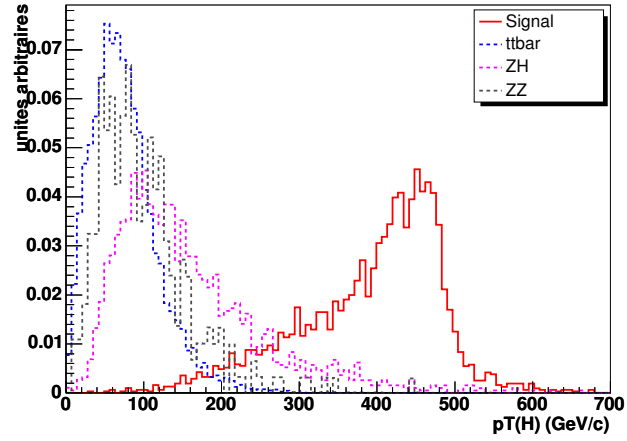


FIGURE 3.39: Impulsion transverse du Higgs reconstruit, pour le canal B1, et quelques-uns de ses bruits de fond caractéristiques, pour $M_{Z_H} = 1000$ GeV.

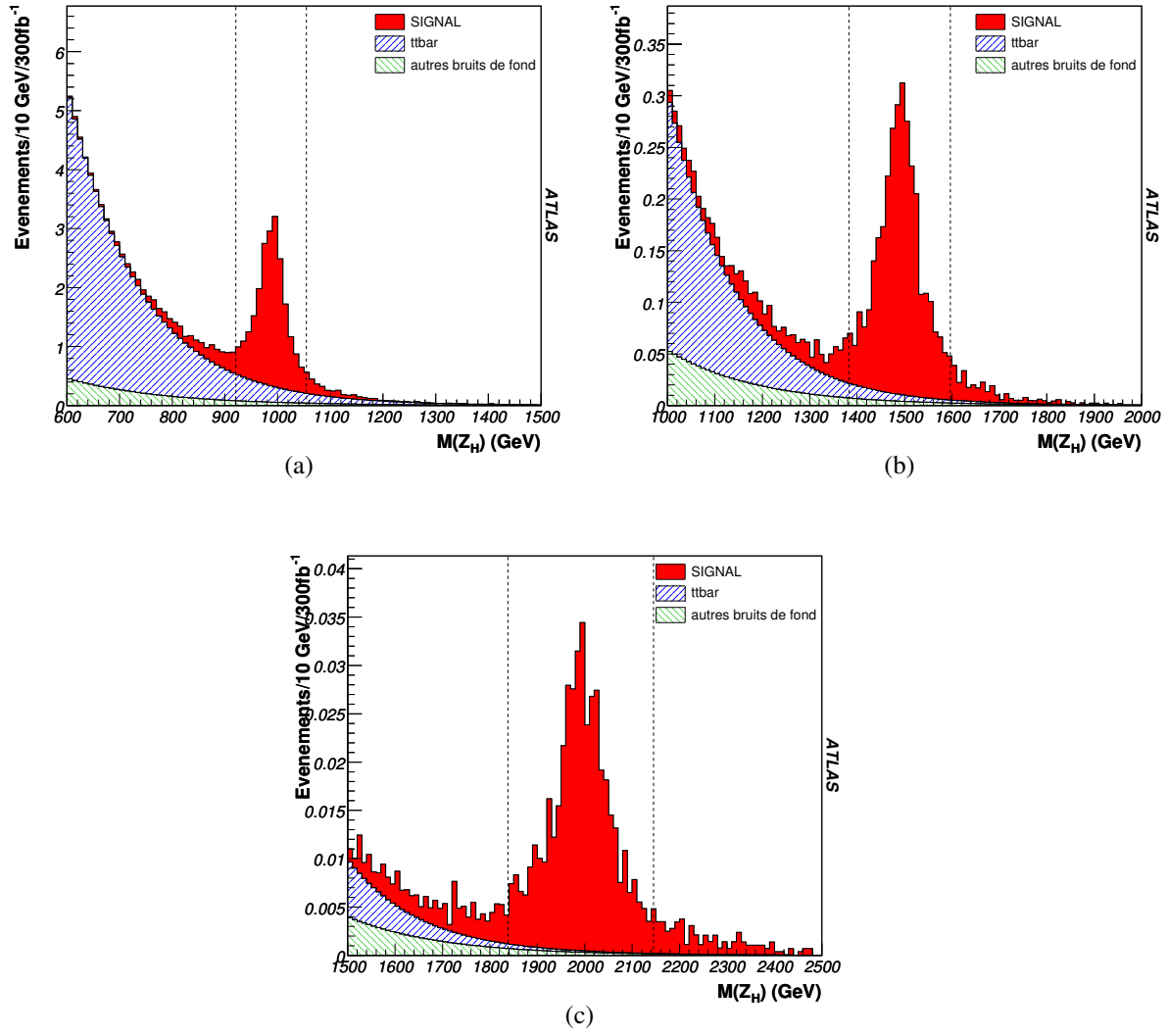


FIGURE 3.40: Masse reconstruite de Z_H dans le canal B1 et de son bruit de fond ($t\bar{t}$ d'une part et tous les autres réunis d'autre part), pour $M(Z_H) = 1000$ (a), 1500 (b) et 2000 (c) GeV, à 300 fb⁻¹.

3.2.5.3 Signal B2

Il s'agit du canal $Z_H \rightarrow Z_1 H \rightarrow \ell^+ \ell^- W_2^+ W_3^- \rightarrow \ell^+ \ell^- jj \ell \nu$. La reconstruction de Z_H est ici très similaire à la méthode employée pour le signal B2, la différence essentielle venant du fait qu'un des 4 leptons est ici un neutrino :

- la première condition est de ne sélectionner que les événements ayant exactement 3 leptons (électrons et muons)
- à partir de ces 3 leptons est formée une paire de leptons de même saveur et de charge opposée, ayant une masse invariante $m(ll)$ telle que $|m(ll) - M_Z| < 5 \text{ GeV}$, de manière à reconstruire le Z primaire qui vient de Z_H
- afin de reconstruire W_2 se désintégrant leptoniquement, le quadri-vecteur du troisième lepton est associée à l'énergie transverse manquante ; l'impulsion longitudinale manquante $p_z(\nu)$ se déduit d'une équation du second degré à laquelle aboutit la condition que la masse invariante de la paire lepton-neutrino doit être égale à M_W :

$$\begin{cases} M_W^2 = \tilde{p}(l) \cdot \tilde{p}(\nu) & \Rightarrow & M_W^2 = [E(l) + E(\nu)]^2 - [\vec{p}_T(l) + \vec{p}_T(\nu)]^2 - [p_z(l) - p_z(\nu)]^2 \\ M_\nu \equiv 0 & \Rightarrow & E^2(\nu) = \vec{p}_T^2(\nu) + p_z^2(\nu) \end{cases}$$

$$\Rightarrow \text{équation du second degré en } p_z(\nu) \quad \left\{ \begin{array}{l} Ap_z^2(\nu) + Bp_z(\nu) + C = 0 \\ \text{avec} \quad \begin{cases} A = \vec{p}_T^2(l) \\ B = [2\vec{p}_T(l) \cdot \vec{p}_T(\nu) + M_W^2]p_z(l) \\ C = [p_x(l)p_y(\nu) - p_y(l)p_x(\nu)]^2 + p_z^2(l)\vec{p}_T^2(\nu) \\ \quad - M_W^4/4 - M_W^2\vec{p}_T(l) \cdot \vec{p}_T(\nu) \end{cases} \end{array} \right.$$

si l'équation n'a pas de solution, alors l'hypothèse alternative que le lepton et son neutrino associé sont colinéaires permet d'obtenir une solution dans tous les cas (cette hypothèse est classique dans le cas d'une désintégration en $\tau \rightarrow \nu X$, elle ne l'est pas pour la désintégration leptonique d'un W ; c'est la grande impulsion du W qui la rend possible ici) :

$$\begin{cases} \vec{p}(\nu) = \alpha \vec{p}(l) & \text{avec } \alpha = \frac{|\vec{p}(\nu)|}{|\vec{p}(l)|} \\ |\vec{p}_T(\nu)| = |\vec{p}_T(miss.)| & (= \text{énergie transverse manquante}) \end{cases}$$

$$\Rightarrow \left\{ \begin{array}{l} p_z(\nu) = \frac{|\vec{p}(\nu)|}{|\vec{p}(l)|} p_z(l) \\ \quad = \text{Signe}[p_z(l)] \cdot \frac{|\vec{p}_T(miss.)|}{\left(\frac{|\vec{p}(l)|}{p_z(l)}\right)^2 - 1} \\ p_x(\nu) = \frac{|\vec{p}(\nu)|}{|\vec{p}(l)|} p_x(l) \\ p_y(\nu) = \frac{|\vec{p}(\nu)|}{|\vec{p}(l)|} p_y(l) \end{array} \right. \quad \text{avec } |\vec{p}(\nu)|^2 = |\vec{p}_T(miss.)|^2 + p_z^2(\nu)$$

- W_3 est reconstruit de la même manière que le Z_3 se désintégrant en jet(s) dans le canal B1 : les paires de jets vérifiant $|m(jj) - M_W| < 15 \text{ GeV}$ et le jet d'impulsion transverse maximum sont associés au premier W pour former des Higgs candidats, la solution étant déterminée par le candidat ayant la plus grande impulsion transverse parmi ceux qui ont leur masse telle que $|m(H) - M_H| < 30 \text{ GeV}$
- le Z primaire (Z_1) et le Higgs reconstruits permettent finalement d'obtenir la masse reconstruite de Z_H

La masse reconstruite est reproduite Figure 3.41. Par rapport au signal B1 a été substitué le boson primaire $Z \rightarrow \ell^+ \ell^-$ par $W \rightarrow \ell \nu$, d'où un bruit de fond un peu plus important et une résolution en masse un peu moins bonne.

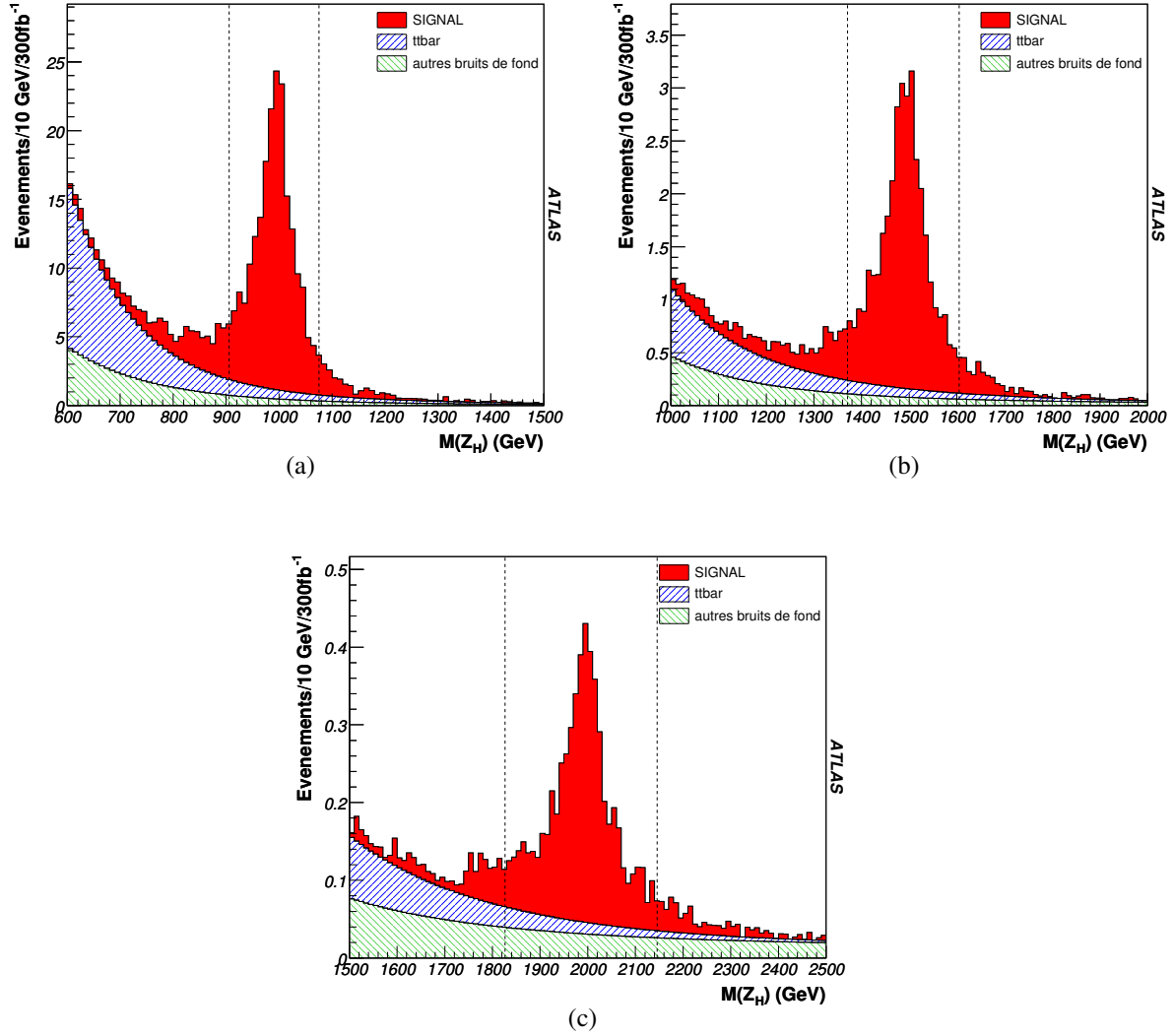


FIGURE 3.41: Masse reconstruite de Z_H du SIGNAL 2 et de son bruit de fond ($t\bar{t}$ d'une part et tous les autres réunis d'autre part), pour $M(Z_H) = 1000$ (a), 1500 (b) et 2000 (c) GeV, à 300 fb^{-1}

3.2.5.4 Signal B3

Il s'agit du mode $W_H \rightarrow W_1 H \rightarrow \ell \nu Z_2 Z_3 \rightarrow \ell \nu \ell^+ \ell^- j j$. Les particules de l'état final sont les mêmes que pour le mode B2, la différence étant que la désintégration leptonique du W vient du W_1 primaire plutôt que du W_2 secondaire. L'analyse est très similaire au canal B2. Les masses invariantes sont représentées sur la Figure 3.42.

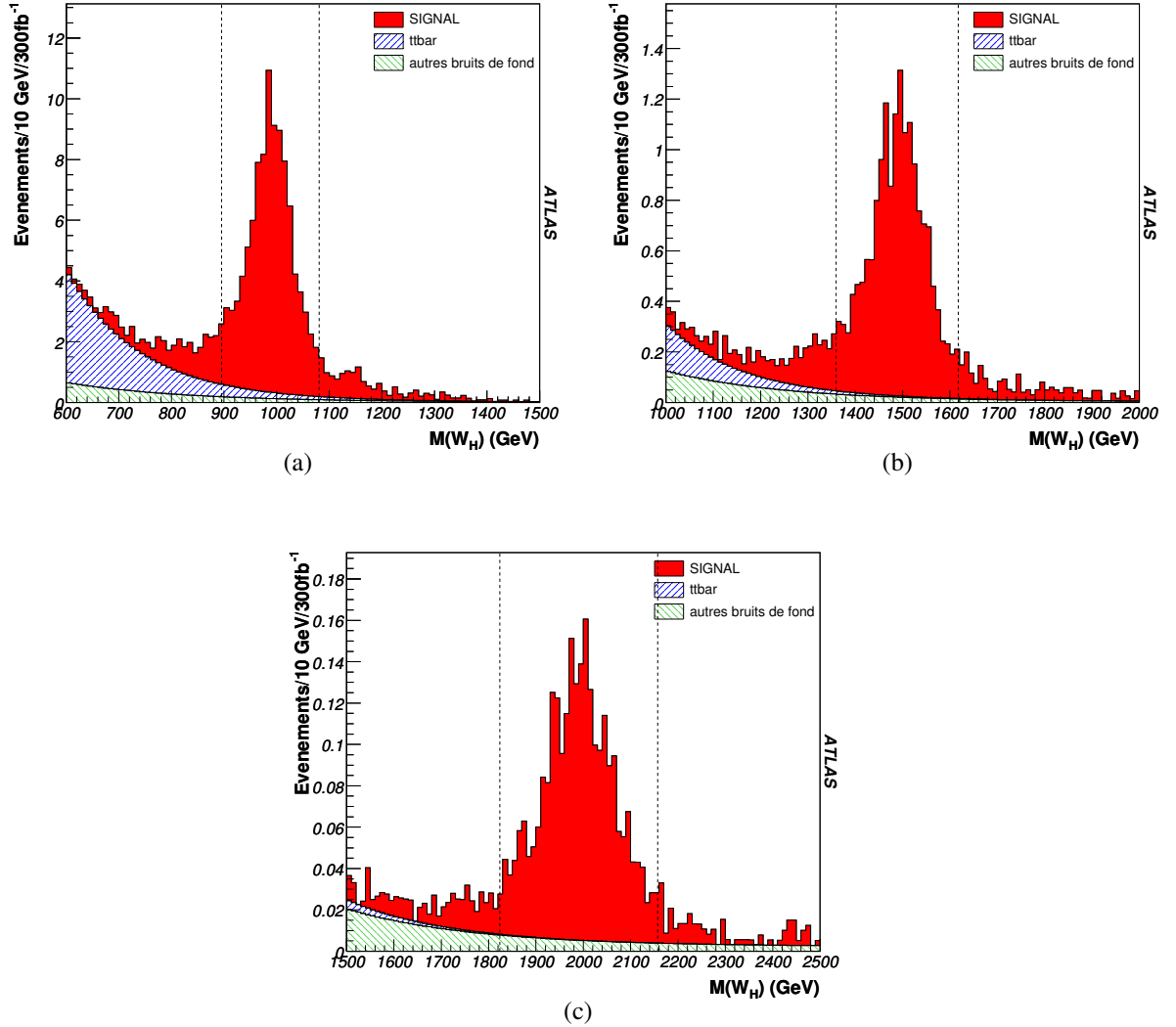


FIGURE 3.42: Masse reconstruite de W_H du SIGNAL 3 et de son bruit de fond ($t\bar{t}$ d'une part et tous les autres réunis d'autre part), pour $M(W_H) = 1000$ (a), 1500 (b) et 2000 (c) GeV, à 300 fb^{-1}

3.2.5.5 Signaux de type A

Il s'agit des signaux A1 $Z_H \rightarrow ZH \rightarrow \ell^+\ell^-W^+W^- \rightarrow \ell^+\ell^- \ell^+\nu\ell^-\nu$ A2 $W_H \rightarrow WH \rightarrow \ell\nu W^+W^- \rightarrow \ell\nu \ell^+\nu\ell^-\nu$ A3 $Z_H \rightarrow ZH \rightarrow jjZZ \rightarrow jj \ell^+\ell^-\ell^+\ell^-$ et A4 $W_H \rightarrow WH \rightarrow jjZZ \rightarrow jj \ell^+\ell^-\ell^+\ell^-$. Les signaux de type A (étudiés à l'université de Valence) sont plus faciles à mettre en évidence car les produits finaux de la désintégration du boson de Higgs ne sont que leptoniques. Les signaux A3 et A4 sont reconstruits simultanément étant donné la dégénérescence en masse des Z_H et W_H et la non séparabilité des Z et W lorsqu'ils sont reconstruit dans le mode hadronique. Le bruit de fond du Modèle Standard est considéré ici comme négligeable étant donné la signature claire et les coupures appliquées. Ces coupures sont :

- coupures cinématiques sur les leptons, et notamment sur leur isolation par paires
- coupures sur les neutrinos, $E_T > 50$ GeV pour le canal A1 et $E_T > 100$ GeV pour le canal A2
- coupures sur les masses invariantes et les impulsions transverses des Z et W , du Higgs, et de Z_H et W_H

Le nombre d'événements de signal reconstruit est indiqué dans le Tableau 3.10. Les masses reconstruites pour $m_{V_H} = 1$ TeV sont présentées sur la Figure 3.43.

En prenant pour hypothèse habituelle une luminosité intégrée de 300 fb^{-1} , 140 événements de signal sont attendus à 1000 GeV, ce qui est favorable à une observation de Z_H et W_H , contrairement aux cas de masse plus élevée avec par exemple moins de 10 événements cumulés à 2000 GeV.

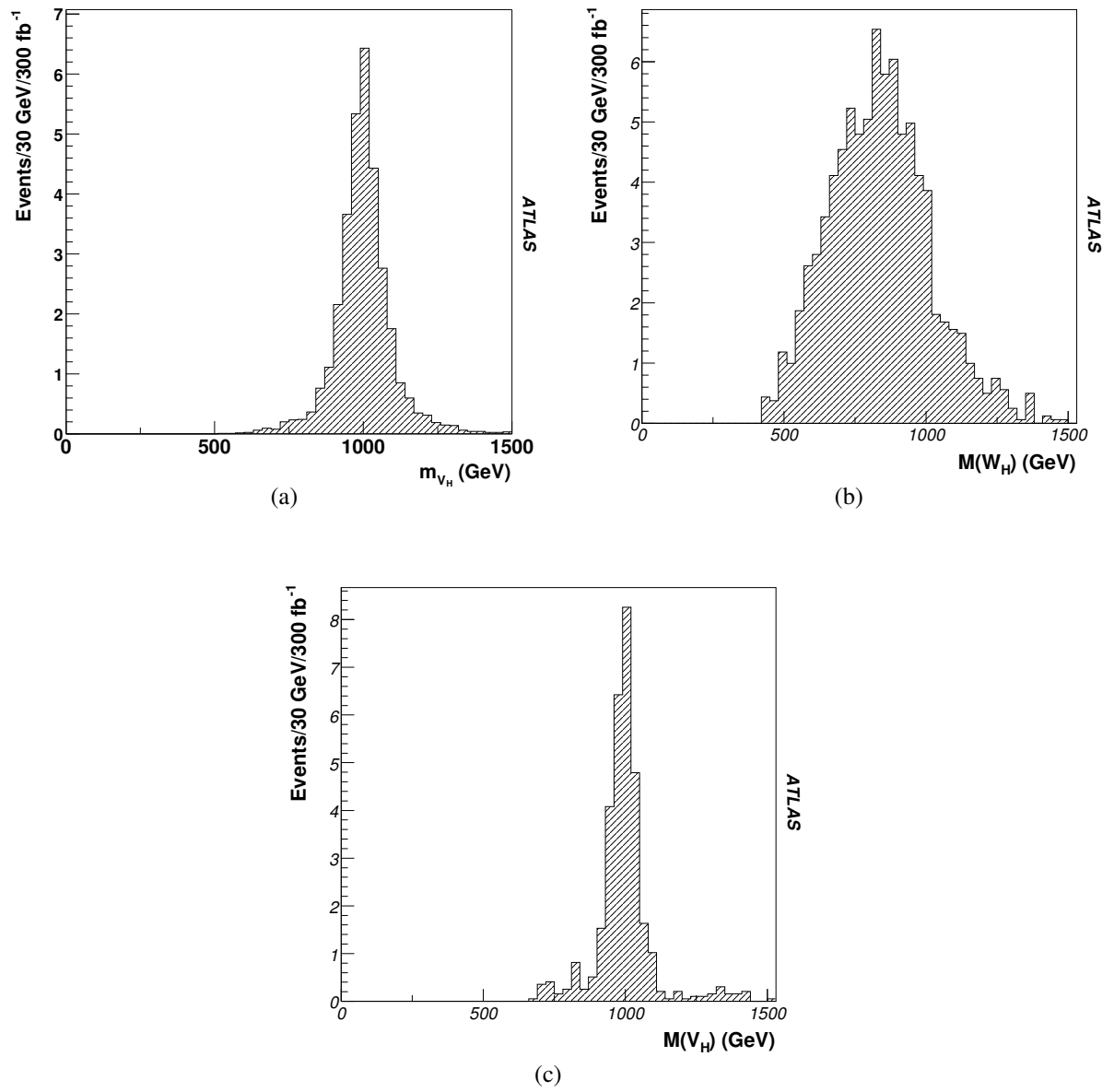


FIGURE 3.43: Masse reconstruite de V_H du signal pour $M_{V_H} = 1000$ GeV et 300 fb^{-1} , pour les signaux A1 (a), A2 (b) et A3/A4 (c).

3.2.5.6 Observabilités

Le Tableau 3.10 résume le potentiel de tous les canaux étudiés pour $m_H = 200$ GeV.

Mode masse (TeV)	A1		A2		A3/A4		B1			B2			B3		
	1	2	1	2	1	2	1	1.5	2	1	1.5	2	1	1.5	2
Signal	32	1.1	77	2.6	29	1.1	18	2.9	0.4	181	31	4.3	92	16	2.4
Bruit de Fond	-	-	-	-	-	-	4.6	0.3	0.02	21	3.9	1.5	6.6	0.7	0.2
Effi (%)	34	23	12	8.0	26	21	17	13	8.7	22	18	12	15	12	7.4
Signification	-	-	-	-	-	-	5.7	2.2	0	23	9.2	2.1	19	7.9	1.8
$(\sigma BR)_{min}(\text{fb})$	43	64	40	60	153	190	170	151	226	29	18	19	70	52	84

TABLE 3.10: Chiffres caractéristiques pour tous les canaux étudiés pour $m_H = 200$ GeV et une luminosité de 300 fb^{-1} . Voir la liste des modes dans le Tableau 3.8 : l'état final ZH est recherché dans les modes A1, A3, B1 et B2, l'état final WH est recherché dans les modes A2, A4 et B3, sachant que les modes A3 et A4 ne peuvent pas être distingués. La dernière ligne est la valeur minimale du produit section efficace par rapport d'embranchement $B(V_H \rightarrow VH)$ autorisant un signal à plus de 5σ et de plus de 10 événements.

Le potentiel de ces analyses a été combiné séparément pour les modes A et les modes B et est résumé Figure 3.45 pour une luminosité de 300 fb^{-1} . Le potentiel de découverte est similaire au cas $m_H = 120$ GeV, alors la limite supérieure sur la masse des bosons de jauge lourd est passée de 2 TeV à 6 TeV, si bien qu'une fraction plus faible de l'espace des paramètres est couverte.

Le potentiel pour une luminosité inférieure peut être évalué approximativement en utilisant le fait que la section efficace augmente d'environ un facteur deux si la masse des bosons V_H décroît de 250 GeV (voir Tableau 3.2). D'autre part, si la luminosité est plus faible tous les modes sauf $W_H \rightarrow W(qq)H(bb)$ sont rapidement limités par la contrainte d'avoir au moins 10 événements de signal, si bien que le potentiel décroît linéairement avec la luminosité. Donc pour une luminosité totale de 75 fb^{-1} , les courbes de Figure 3.45 doivent être décalées vers la gauche d'environ 500 GeV, sauf pour $W_H \rightarrow W(qq)H(bb)$ qui doit être décalées de 250 GeV environ.

3.2.6 Recherche des désintégrations hadroniques des Z_H and W_H

Si les modes de désintégrations leptoniques des Z_H et W_H permettent une découverte plus rapide des bosons de jauge lourd, le test des modèles du petit Higgs, demande de mesurer d'autres modes de désintégrations comme WH or ZH , mais aussi les modes hadroniques [22]. En particulier, pour $\cot \theta \sim 1$, les désintégrations $B(W_H \rightarrow WH)$ and $B(Z_H \rightarrow ZH)$ disparaissent, et les rapports d'embranchement vers les quarks lourds sont [16] :

$$B(Z_H \rightarrow b\bar{b}) = B(Z_H \rightarrow t\bar{t}) = 1/8 = 12.5\% \quad (3.11)$$

$$B(W_H \rightarrow tb) = 1/4 = 25\% \quad (3.12)$$

L'observabilité de ces trois états finaux a été étudiée en simulation rapide. L'efficacité de l'étiquetage des b a été calibrée en simulation complète en particulier pour les jets de très haut p_T ,

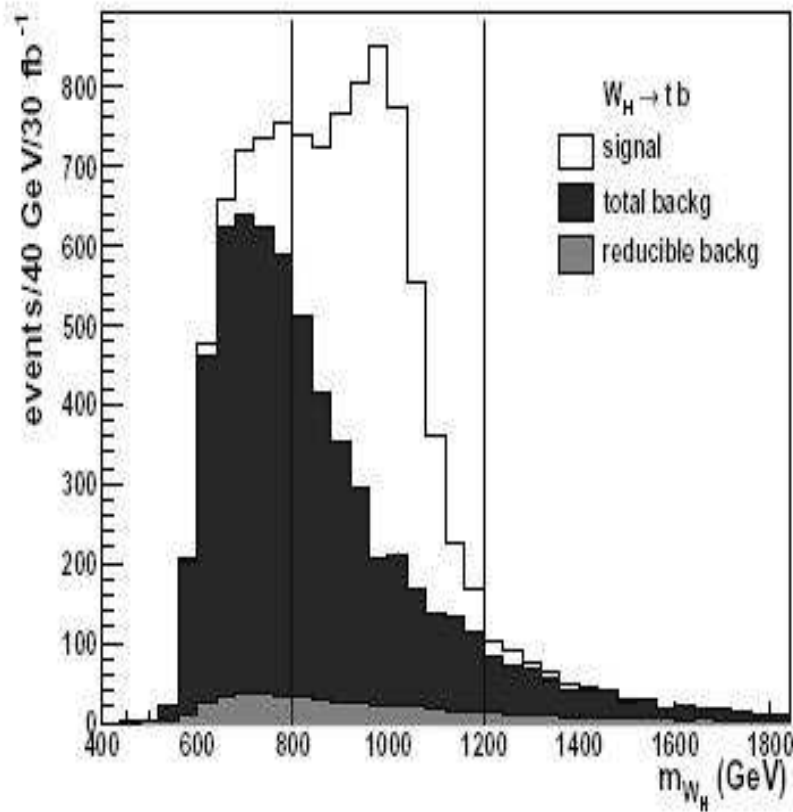


FIGURE 3.44: Masse reconstruite tb pour $M_{W_H} = 1$ TeV et $\cot \theta = 1$.

jusqu'à 1 TeV. Si aucun signal convaincant ne peut être observé pour le Z_H , la désintégration $W_H \rightarrow tb$ peut être visible, comme indiqué ci-dessous.

Le top est reconstruit dans l'état final $W(\ell\nu)b$. Les éléments suivants sont recherchés : un lepton de haut pT , deux b -jet identifiés, dont l'un est proche du lepton, l'autre recule contre le lepton. Le quadrivecteur du neutrino est estimé à partir de de l'énergie transverse manquante et en supposant qu'il est parallèle au lepton. L'efficacité de reconstruction est de 25% et la résolution en masse de 110 GeV pour $M_{W_H} = 1$ TeV. Le bruit de fond vient principalement des événements $t\bar{t}$ et $W + jets$.

Le spectre de masse reconstruit apparaît Figure 3.44 : le signal est clairement visible. Figure 3.45 montre la zone accessible : la région autour de $\cot \theta = 1$ qui n'était pas couverte par les analyses $W_H(Z_H) \rightarrow W(Z)H$ est couverte jusqu'à $M_{W_H} = 2.5$ TeV.

3.2.7 Généralisation

Les recherches de bosons lourds exposés dans les pages précédentes se placent dans le cadre d'un modèle bien précis du petit Higgs : le potentiel dans l'espace des paramètres est indiqué Figure 3.45. Ces recherches peuvent être aussi présentées autant que possible de façon modèle indépendante pour permettre une interprétation dans d'autres modèles. Le fait que les résonances soient entièrement reconstruites et relativement étroites (même en présence de neutrinos) sur un bruit de fond modéré rend ces analyses robustes envers un changement de modèle.

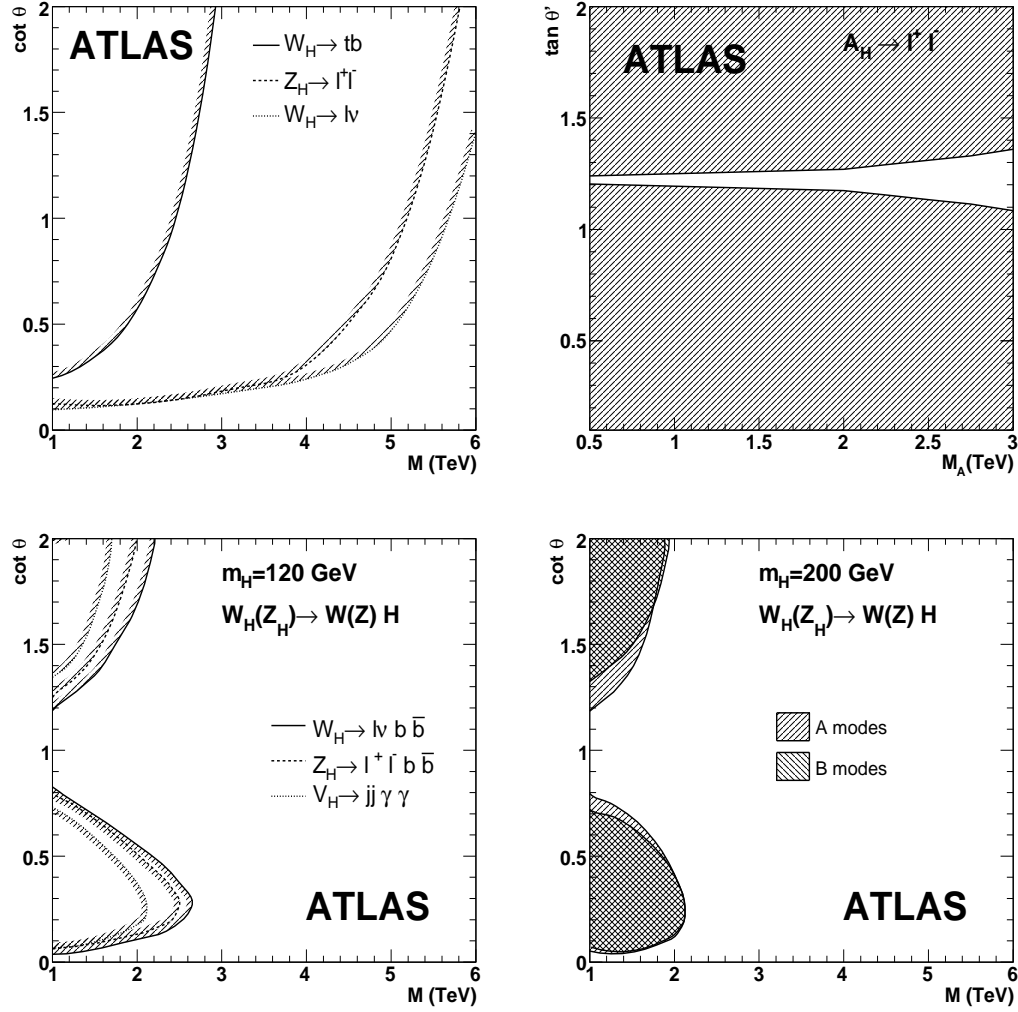


FIGURE 3.45: Les régions accessibles à une découverte à 5σ des bosons de jauge lourd W_H , Z_H and A_H en fonction de leur masse, et de $\cot \theta$ ou $\tan \theta'$. Les régions à gauche des lignes sont accessibles avec 300 fb^{-1} : en haut à gauche pour $A_H \rightarrow e^+ e^-$, en haut à droite pour les désintégrations leptoniques ou hadroniques de W_H ou Z_H , en bas à gauche pour les désintégrations avec un boson de Higgs dans l'état final dans l'hypothèse $m_H = 120$ GeV (pour laquelle $m_{V_H} < 2$ TeV), en bas à droite pour les désintégrations avec un boson de Higgs dans l'état final dans l'hypothèse $m_H = 200$ GeV (pour laquelle $m_{V_H} < 6$ TeV).

Figure 3.46 indique pour une observation à 5σ d'une résonance V_H la valeur minimale du produit section efficace de V_H par le rapport d'embranchement $V_H \rightarrow VH$ pour les masses du boson de Higgs de 120 GeV et 200 GeV. Ces courbes sont dérivées directement de la recherche $Z_H \rightarrow ZH$ et $W_H \rightarrow WH$ résumée dans les Tableau 3.10 et s'appliquent par exemple au A_H et à toute autre particule sous les hypothèses suivantes :

- largeur de la résonance R petite devant la largeur expérimentale $\sigma_{exp} \simeq 50$ GeV à 1 TeV augmentant à 80 GeV à 2 TeV pour tous les modes.
- absence de bruit de fond spécifique au modèle. Par exemple dans le cas du A_H : la masse du A_H doit être significativement différente de la masse des W_H et Z_H .
- la résonance recherchée ayant une masse de l'ordre du TeV voire plus, l'efficacité de sélection des coupures en acceptance (en pseudo-rapacité comme en impulsion minimum des produits de désintégration) dépend peu du détail du mécanisme de production (production centrale) et de désintégration (pas de sélection explicite ni implicite sur les angles de désintégration). Toutefois, l'hypothèse implicite est que la résonance R soit produite seule au seuil, donc avec une impulsion transverse et longitudinale de l'ordre de sa masse, et non pas accompagnée d'une autre particule lourde, ni issue d'une particule encore plus lourde

Le boson de Higgs est supposé se désintégrer comme dans le modèle standard, voir Tableau 3.3. Si la masse du boson de Higgs ne correspond pas aux hypothèses 120 GeV ni 200 GeV, les courbes peuvent être corrigées pour le changement du rapport d'embranchement des différents modes. Toutefois les modes impliquant $H \rightarrow ZZ$ et $H \rightarrow WW$ verront une augmentation du bruit de fond si $m_H < 2*m_Z$ et $2*m_W$ respectivement car l'un des bosons vecteurs serait alors en dessous de sa couche de masse. De même, une modification des rapports d'embranchement du boson de Higgs par rapport au modèle standard peut être prise en compte facilement. Pour $m_H = 120$ GeV les modes impliquant $H \rightarrow \gamma\gamma$ sont de 5 à 2 fois moins sensibles que les modes impliquant $H \rightarrow bb$ mais le bruit de fond est plus facilement contrôlable car le $H \rightarrow \gamma\gamma$ sera visible de façon inclusive. Pour $m_H = 200$ GeV, le mode B2 impliquant le Z_H et les modes A2 et B3 impliquant le W_H sont les plus sensibles. La notion de T-parité a été introduite récemment dans certains modèles de "Petit Higgs" (voir par exemple Ref [28]) pour permettre de limiter la contrainte sur l'échelle de masse venant des mesures électrofaibles. La T-parité joue un rôle analogue à la R-parité de SuSy, obligeant la production des nouvelles particules par paires. Dans ce cas les bosons lourds seraient produits par paires, $W_H Z_H$ et $W_H W_H$ étant les modes les plus abondants, avec des sections efficaces pouvant atteindre 100 fb. Comme indiqué ci-dessus, les limites données Figure 3.46 ne peuvent pas s'appliquer directement à ces nouveaux états finaux, mais donnent des pistes sur les modes à étudier en priorité.

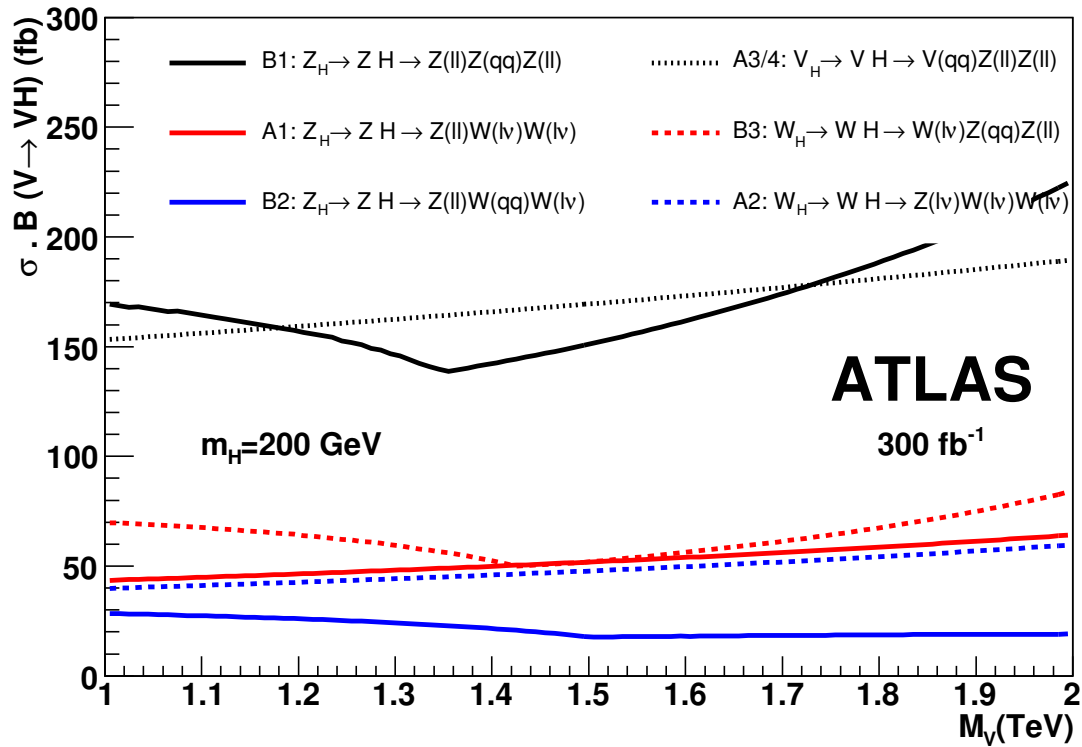
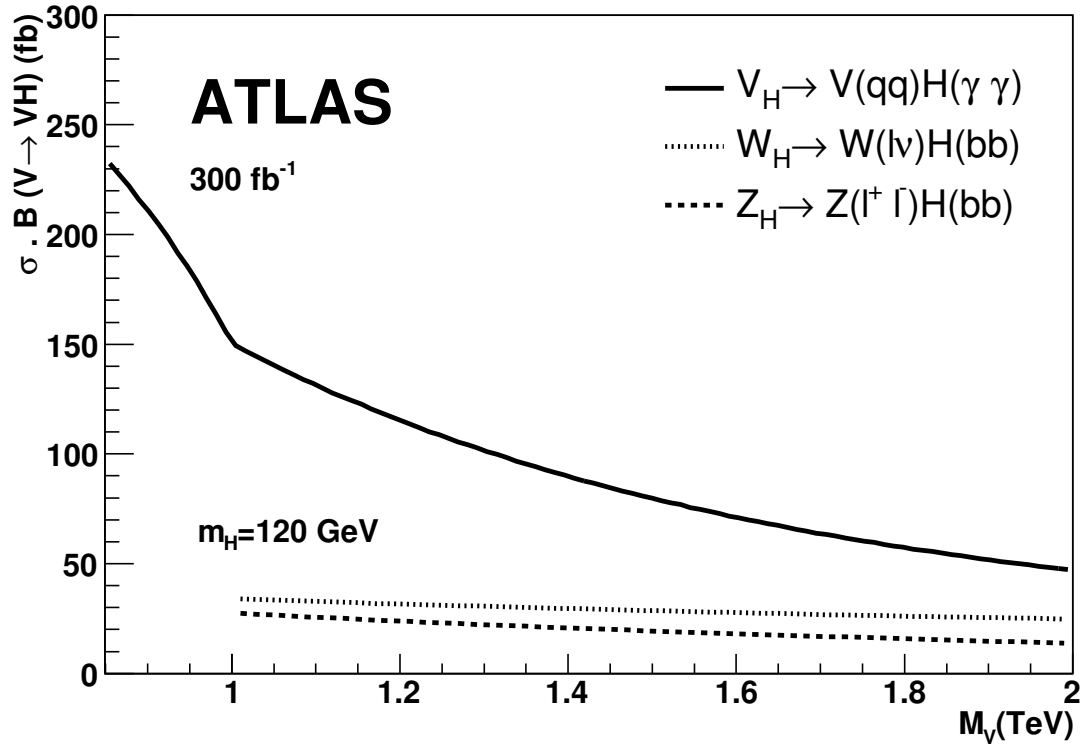


FIGURE 3.46: Graphe en fonction de la masse du minimum du produit section efficace par rapport d'embranchement $B(V_H \rightarrow VH)$ permettant la découverte avec une luminosité de 300 fb⁻¹ d'une résonance V_H . En haut pour $m_H = 120$ GeV dans les canaux $V_H \rightarrow ZH \rightarrow \ell\bar{\ell}b\bar{b}$, $V_H \rightarrow ZH \rightarrow qq\gamma\gamma$ et $V_H \rightarrow WH \rightarrow \ell\nu b\bar{b}$; en bas pour $m_H = 200$ GeV dans les modes étudiés : en ligne pleine pour les états finaux ZH (B1, A1 et B2), en tiretés pour les états finaux WH (B3 et A2) et en pointillé pour les modes A3 et A4 où les états finaux ne peuvent pas être séparés.

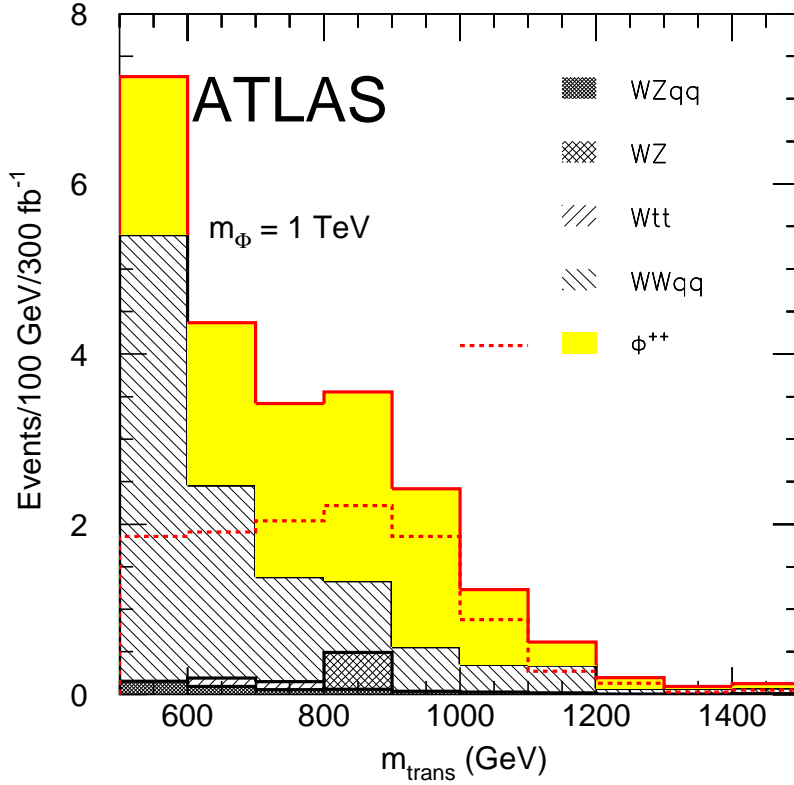


FIGURE 3.47: Distribution de la variable de masse m_{trans} (voir texte) pour un ϕ^{++} de masse 1 TeV et pour $v' = 25$ GeV. L'histogramme tireté montre le signal seul et l'histogramme plein montre la somme signal et du bruit de fond. Les composantes du bruit de fond sont montrés séparément.

3.3 Recherche de ϕ^{++}

Le boson de Higgs doublement chargé peut être produit par paire et désintégration dans les modes leptoniques via $q\bar{q} \rightarrow \phi^{++}\phi^{--} \rightarrow llll$, mais la section efficace pour ce processus est très faible au LHC. Le processus de fusion WW du type $dd \rightarrow uu\phi^{++} \rightarrow uuW^+W^+$ est plus envisageable. Cela donne des événements avec deux leptons de même charge et de l'énergie manquante venant des désintégrations des W . Le couplage $\phi^{++}WW$ est déterminé par v' , l'énergie moyenne dans le vide (vev) de l'élément neutre du triplet. La sensibilité des recherches au LHC a été étudiée en fonction de v' et de la masse du ϕ^{++} . Pour $v' = 25$ GeV et une masse de 1 TeV, le taux de production de ϕ^{++} suivi par la désintégration de WW est 4.9 fb en imposant au W $|\eta| < 3$ et $p_T > 200$ GeV [16]. Comme dans le cas de la recherche du Higgs du modèle standard dans les processus de fusion de vecteur ("VBF"), la présence de deux jets opposés à grande rapidité peut être utilisée pour réduire le bruit de fond. La sélection des événements est très proche de ce qui est fait pour la recherche de Higgs du modèle standard produit par VBF [20, 29] :

- deux leptons de même signe (électrons ou muons) avec $|\eta| < 2.5$.
- deux leptons de plus de $p_T > 20$ GeV, dont l'un de plus de $p_T > 150$ GeV (ce qui permet d'assurer le déclenchement)
- les impulsions des leptons ne doivent pas se compenser : $|p_{T1} - p_{T2}| > 200$ GeV.

- la différence en pseudo-rapacité des deux leptons doit vérifier $|\eta_1 - \eta_2| < 2$.
- $E_T^{miss} > 50$ GeV.
- deux jets avec $p_T > 15$ GeV, de pseudorapacité opposée, séparé en pseudorapacité par $|\eta_1 - \eta_2| > 5$; un jet doit avoir $E > 200$ GeV et l'autre $E > 100$ GeV.

La masse invariante du système WW ne peut pas être reconstruite mais le signal peut être mis en évidence avec la variable m_{trans} construite avec les impulsions des leptons (\mathbf{p}_1 and \mathbf{p}_2) et l'impulsion manquante transverse \mathbf{p}_T^{miss} :

$$m_{trans}^2 = (E_1 + E_2 + |E_T^{miss}|)^2 - (\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_T^{miss})^2 \quad (3.13)$$

La distribution de cette variable de masse reconstruite apparaît Figure 3.47 pour une masse de 1 TeV. Les bruits de fond du modèle standard sont montrés séparément. Le signal est visible mais avec un taux très faible. La mise en évidence de ce processus exige une très grande luminosité intégrée, la possibilité de détecter les jets à grande pseudorapacité, et la possibilité de contrôler les bruits de fond. En demandant au moins 10 événements avec $m_{trans} > 700$ (resp. 1000) GeV pour $M_\phi = 1000$ (resp. 1500) GeV et une valeur de $S/\sqrt{B} > 5$ rend possible la découverte pour $v' > 29$ (resp. 54) GeV. Ces valeurs sont supérieures à la contrainte $v' < 25$ GeV venant de l'ajustement des mesures électrofaibles [16], ce qui rend peu prometteuse cette recherche.

La production par paire $q\bar{q} \rightarrow \phi^{++}\phi^{--} \rightarrow llll$ a été étudié à CMS [30] : ce signal serait visible à 5σ jusqu'à une masse de 650 GeV.

3.4 Contraintes sur le modèle, conclusion et perspectives

Nous avons montré par une série d'analyses choisie comment les mesures avec le détecteur Atlas au LHC peut être utilisées pour mettre en évidence les différentes particules prédites par le modèle du petit Higgs avec une luminosité intégrée de 300 fb^{-1} .

Le quark lourd T est observable jusqu'à une masse d'environ 2.5 TeV dans la désintégration Wb . La sensibilité des modes Zt et Ht est plus limitée mais couvre néanmoins une bonne partie du domaine autorisé pourvu que la masse du boson de Higgs ne soit pas trop grande. Le canal $H \rightarrow b\bar{b}$ est utile jusqu'à une masse du boson de Higgs d'environ 150 GeV. Dans ce cas, Atlas pourra détecter le quark lourd T dans trois canaux de désintégration permettant de tester le modèle.

En ce qui concerne les nouveaux bosons de jauge lourd, la situation est résumée Figure 3.45, montrant les régions accessibles dans les états finaux leptoniques des Z_H et W_H en fonction de l'angle de mélange. Toutefois, de telles désintégrations sont prévues dans de nombreux modèles au-delà du modèle standard, leur observation n'est donc pas suffisante pour prouver la validité du modèle de petit Higgs. Figure 3.45 montre également la région accessible pour l'observabilité des désintégrations Z_H et W_H dans différents états finaux en fonction de $\cot\theta$ et des masses. Certains modes de désintégrations ne sont visibles que pour les masses de boson faibles et en dehors de la région $\cot\theta \simeq 1$. Cette région peut être couverte par la désintégration $W_H \rightarrow tb$.

Le cas du boson de Higgs doublement chargé ϕ^{++} est moins prometteur. Le secteur du Higgs est moins contraint par les arguments de réglage fin si bien que la masse du ϕ^{++} peut aller jusqu'à 10 TeV. La sensibilité d'Atlas n'atteint que 2 TeV et encore seulement pour des grandes valeurs de v' . D'autres modèles du Petit Higgs ont une structure différente du secteur du Higgs, avec en particulier plus d'un doublet de Higgs. Ces modèles n'ont pas encore été étudié. Le

domaine du "Petit Higgs" voit une grande activité du point de vue théorique si bien que dans le cas des modes ZH et WH les limites sur les différents modes ont été données de façon modèle indépendante sur la Figure 3.46 pour pouvoir être exploitées dans d'autres modèles.

Même si cette analyse est plus détaillée qu'une étude de faisabilité, plusieurs points devront être améliorés dans le futur :

- les analyses devront être réitérées en simulation complète en particulier en ce qui concerne la reconstruction des masses avec des jets à très haute énergie, et l'identification des leptons proches des jets. Une étude préliminaire sur les bosons lourds [9] a montré une perte d'efficacité d'un facteur deux, ce qui est significatif sans être dramatique, car cela a été obtenu sans réajustement de la reconstruction qui n'a pas été particulièrement réglée pour la reconstruction et l'identification de jets et leptons de très haute énergie.
- les bruits de fond ont été générés avec Pythia. Des nouveaux générateurs à l'ordre suivant (Next-to-Leading-Order) sont disponibles et en plein développements depuis quelques années (Alpgen, MC@NLO), et permettent une meilleure description des données Tevatron (quoique pas encore idéale), en particulier à haute impulsion. Notons néanmoins que les signaux reconstruits présentés dans ce document sont visible sous forme d'une résonance sur un bruit de fond relativement plat, et sont donc pas très sensible au niveau de bruit de fond.

Les analyses présentées prenant tout leur sens après plusieurs années de fonctionnement du LHC, la reconstruction à haute énergie aura pu être réglée, et les Monte-Carlo ajustés sur les données elle-mêmes ; de plus le boson de Higgs aura été découvert, sa masse et une partie de ses rapports d'embranchement mesurés, voire même les bosons lourds dans leurs modes de désintégration leptoniques.

CONCLUSION

Une riche bibliothèque d'algorithmes de reconstruction a été développée, dans un environnement qui s'est peu à peu complexifié. Le code de reconstruction, initialement conçu sur une simulation Monte-Carlo plutôt optimiste, est en train d'être confronté au réalisme d'une simulation fidèle à la construction effective du détecteur. Après l'expérience des tests en faisceau combinés, des rayons cosmiques il va connaître les événements du halo du faisceau et des interaction faisceau-gas, avant les premières collisions à 900 GeV fin 2007, avant le début de la prise de données de physique milieu 2008. Le défi sera de conserver l'équilibre entre un code robuste à fonctionnalité minimale pour analyser les premières données et effectuer les premiers étalonnages dès le premier jour, et la mise au point d'algorithmes sophistiqués nécessaires pour «le jour d'après». La stabilisation de la définition des données persistantes connaîtra le même dilemme entre l'aspect «proche du détecteur» et l'aspect «proche de la physique». Le «Petit Higgs» est une théorie effective qui peut être prise au premier degré, ou bien comme prétexte à l'étude de résonances lourdes. Dans plusieurs cas ces résonances peuvent effectivement être détectées avec une luminosité importante, dans des canaux originaux comme tZ , ZH ou WH . Avant de traiter le boson de Higgs dans l'état final au même titre que top, Z ou W, il ne manque qu'un détail : il faut d'abord l'avoir découvert...mais ceci est une autre histoire.

GLOSSAIRE

AOD : Analysis Object Data : Données persistantes préparées pour l'analyse
ATNIGHT : logiciel gérant la reconstruction chaque nuit de l'intégralité des logiciels d'Atlas
Athena : framework de tous les logiciels d'atlas
CBNT : ComBined NTuple : ntuple commun concernant le résultat de la reconstruction
CMT : Configuration Management Tool : logiciel permettant de gérer en particulier la compilation et de l'ensemble des logiciels d'Atlas
CSC : Cathode Strip Chamber : détecteur dans les chambres à muons
CVS : Concurrent Versioning System : logiciel permettant de gérer l'évolution de tout ensemble de fichiers (en format text en général), et en particulier l'intégralité du code source des logiciels d'Atlas.
EDM : Event Data Model : ensemble des classes représentant les données d'Atlas au cours de leur transformations successives
ESD : Event Summary Data : Données persistantes issues de la reconstruction
HITS : Données persistantes issues de la simulation Geant4
HepMC : composante de l'EDM contenant toutes les particules "vraies" issues de la simulation Monte-Carlo
MDT : Muon Drift Tube : détecteur dans les chambres à muons
NICOS : logiciel de test automatiques interactifs
Pixel : première partie du trajectographe, composé de plaquettes de Silicium divisées en pixels
RDO : Raw Data Object : Données persistantes issues de la digitisation ou bien d'ATLAS
ROD : Read Out Driver : composant électronique assurant le dernier traitement des données en-ligne de chaque détecteur
ROI : Region Of Interest : région de l'angle solide d'Atlas que le déclenchement aura repéré
RPC : Resistive Plate Chamber : détecteur dans les chambres à muons
RTT : Run Time Testing : logiciels de test automatiques en batch
SCT : SiliCon Tracker : deuxième partie du trajectographe, composé de plaquette de Silicium lues par des rubans
StoreGate : partie de Athena gérant le TES et le TDS
TAG : Données persistantes permettant une préselection rapide des événements
TDR : Technical Design Report : documents publiés à la fin des années de 1990 donnant les caractéristiques et performances attendues de tous les détecteurs.
TDS : Transient Detector Store : espace de mémoire virtuel où sont stockés les objets ne changeant pas (ou rarement) d'un événement à l'autre
TES : Transient Event Store : espace de mémoire virtuel où sont stockés les objets de l'EDM au cours de l'exécution des logiciels d'Atlas
TGC : Thin Gap Chamber : détecteur dans les chambres à muons
TRT : Transition Radiation Tracker : troisième partie du trajectographe, composé de pailles dans une matrice favorisant l'émission de photon de transition par les électrons
XML : Format ascii de fichier texte (sur lequel html est basé)

RÉFÉRENCES

- [1] ATLAS, Physics and detector performance technical design report, 1999, LHCC 99-14/15.
- [2] M. Cattaneo *et al.*, Proceedings of 2001 conference for computing in high energy and nuclear physics, beijing, 2001.
- [3] V. Boisvert *et al.*, Final report of the reconstruction task force, 2003, ATLAS-SOFT-2003-010.
- [4] J. C. Solveig Albrand and J. Fulachier, Proceedings of 2003 conference for computing in high-energy and nuclear physics (chep 03), la jolla, california, 2003.
- [5] J. Bénard *et al.*, *L'eXtreme Programming* (Eyrolles, 2002).
- [6] S. Albrand *et al.*, Atlas C++ coding standard, 2003, ATLAS-SOFT-2002-001.
- [7] J. Cavarero and R. Lecat, *La conception orientée objet, évidence ou fatalité* (Ellipses, 2002).
- [8] G. Theraulaz *et al.*, The formation of spatial patterns in social insects : from simple behaviours to complex structures, Phil. Trans. R. Soc. Lond. A (2003) **361**, 1263 (2003), [<http://dx.doi.org/10.1098/rsta.2003.1198>].
- [9] M. Lechowski, *Test du modèle du Petit Higgs dans ATLAS au LHC. Simulation de la numérisation du calorimètre électromagnétique*, Thèse de doctorat, Université Paris-Sud 11, 2005.
- [10] <http://wwwinfo.cern.ch/asdoc/hbook.html3/hboomain.htm>.
- [11] <http://root.cern.ch/>.
- [12] <http://pool.cern.ch/>.
- [13] K. Assamagan *et al.*, Final report of the AOD/ESD definition task force, 2004, ATLAS-SOFT-2004-006.
- [14] ATLAS, Atlas computing technical design report, 2005, CERN LHCC 2005-022.
- [15] D. Froidevaux and P. Sphicas, General-purpose detectors for the large hadron collider, Ann. Rev. Nucl. Part. Sci. **56**, 375 (2006).
- [16] T. Han, H. E. Logan, B. McElrath and L.-T. Wang, Phenomenology of the little higgs model, Phys. Rev. **D67**, 095004 (2003), [[hep-ph/0301040](http://arxiv.org/abs/hep-ph/0301040)].
- [17] A. Buras *et al.*, Rare k and b decays in the littlest higgs model without t-parity, J. of High Energy Physics **11**, 062 (2006), [[hep-ph/0607189](http://arxiv.org/abs/hep-ph/0607189)].
- [18] J. Garcia, M. Lechowski, E. Ros and D. Rousseau, Search for the decays $z_h \rightarrow zh$ and $w_h \rightarrow wh$ in the little higgs model assuming $m_h = 120$ gev, 2004, ATLAS-PHYS-2004-001.
- [19] J. Garcia, M. Lechowski, E. Ros and D. Rousseau, Search for the decays $z_h \rightarrow zh$ and $w_h \rightarrow wh$ in the little higgs model assuming $m_h = 200$ gev, 2004, ATLAS-COM-PHYS-2004-032.
- [20] G. Azuelos *et al.*, Exploring little higgs models with atlas at the lhc, Eur. Phys. J. **C39S2**, 13 (2005), [[hep-ph/0402037](http://arxiv.org/abs/hep-ph/0402037)].
- [21] J. E. Garcia-Navarro, *Performance of the ATLAS Silicon Tracker and Search for new Gauge Bosons in the Little Higgs Model*, Tesis doctoral, Universidad de Valencia, 2004.
- [22] S. González de la Hoz, L. March and E. Ros, Search for hadronic decays of Z_H and W_H in the little Higgs model, 2006, ATL-PHYS-PUB-2006-003.
- [23] L. Sjostrand, T. Lonnblad and S. Mrenna, *PYTHIA 6.2 : Physics and manual*, 2001, [hep-ph/0108264](http://arxiv.org/abs/hep-ph/0108264).

- [24] S. Asai *et al.*, Prospects for the search of a standard model higgs boson in atlas using vector boson fusion, Eur. Phys. J **C32**, s19 (2004).
- [25] G. Burdman, M. Perelstein and A. Pierce, Collider tests of the little higgs model, Phys. Rev. Lett. **90**, 241802 (2003), [hep-ph/0212228].
- [26] A. Djouadi, The anatomy of electro-weak symmetry breaking, hep-ph/0503172.
- [27] T. Binoth, A full next-to-leading order study of direct photon pair production in hadronic collisions, Eur. Phys. J. **C16**, 311 (2000).
- [28] A. Belyaev, C.-R. Chen, K. Tobe and C. P. Yuan, Phenomenology of littlest higgs model with t-parity : Including effects of t-odd fermions, Phys. Rev. **D74**, 115020 (2006), [hep-ph/0609179].
- [29] G. Azuelos, K. Benslama and G. Couture, The little higgs : analysis of the channel $w^+w^- \rightarrow \phi^{++} \rightarrow w^+w^+$, 2004, ATL-PHYS-2004-002.
- [30] CMS, CMS physics technical design report, 2006, CERN LHCC 2006-021.